Journal of Applied Computer Science Methods
Published by University of Social Sciences

ACSM

Volume 7 Number 2 2015
University of Social Sciences, IT Institute
Społeczna Akademia Nauk
Łódź
Instytut Technologii Informatycznych
ISSN 1689-9636

International Journal of Applied Computer Science Methods

Journal of Applied Computer Science Methods

Published by University of Social Sciences

ACSM

Volume 7 Number 2 2015

University of Social Sciences, IT Institute

SPOŁECZNA AKADEMIA NAUK
ŁÓDŹ

Instytut Technologii Informatycznych
iti

International Journal of Applied Computer Science Methods (JACSM) is a semi-annual periodical published by the University of Social Sciences (SAN) in Lodz, Poland.

Publishing and Editorial Office:
University of Social Sciences (SAN)
Information Technology Institute (ITI)
Sienkiewicza 9
90-113 Lodz
Tel.: +48 42 6646654
Fax.: +48 42 6366251
E-mail: acsm@spoleczna.pl
URL: http://acsm.spoleczna.pl

Print: Mazowieckie Centrum Poligrafii, ul. Duża 1, 05-270 Marki, www.c-p.com.pl, biuro@c-p.com.pl

---

Aims and Scope:

The **International Journal of Applied Computer Science Methods is** a semi-annual, refereed periodical, publishes articles describing recent contributions in theory, practice and applications of computer science. The broad scope of the journal includes, but is not limited to, the following subject areas:
Knowledge Engineering and Information Management: *Knowledge Processing, Knowledge Representation, Data Mining, Machine Learning, Knowledge-based Systems, Knowledge Elicitation, Knowledge Acquisition, E-learning, Web-intelligence, Collective Intelligence, Language Processing, Approximate Reasoning, Information Archive and Processing, Distributed Information Systems.*
Intelligent Systems: *Intelligent Database Systems, Expert Systems, Decision Support Systems, Intelligent Agent Systems, Artificial Neural Networks, Fuzzy Sets and Systems, Evolutionary Methods and Systems, Hybrid Intelligent Systems, Cognitive Systems, Intelligent Systems and Internet, Complex Adaptive Systems.*
Image Understanding and Processing: *Computer Vision, Image Processing, Computer Graphics, Pattern Recognition, Virtual Reality, Multimedia Systems.*
Computer Modeling, Simulation and Soft Computing: *Applied Computer Modeling and Simulation, Intelligent Computing and Applications, Soft Computing Methods, Intelligent Data Analysis, Parallel Computing, Engineering Algorithms.*
Applied Computer Methods and Computer Technology: *Programming Technology, Database Systems, Computer Networks Technology, Human-computer Interface, Computer Hardware Engineering, Internet Technology, Biocybernetics.*

Distribution:

Apart from the standard way of distribution (in the conventional paper format), on-line dissemination of the JACSM is possible for interested readers.

Contents

Convergence Analysis of Multilayer Feedforward Networks Trained with Penalty terms: A Review

Jian Wang[1], Guoling Yang[1], Shan Liu[1], Jacek M. Zurada[2,3]

[1] College of Science
China University of Petroleum, Qingdao, Shandong, China
wangjiannl@upc.edu.cn
yangguolingfwz@163.com
liushan@upc.edu.cn

[2] Department of Electrical and Computer Engineering
University of Louisville, Louisville, Kentucky, USA

[3] Information Technology Institute, University of Social Sciences, Łódź, Poland
jacek.zurada@louisville.edu

Abstract

Gradient descent method is one of the popular methods to train feedforward neural networks. Batch and incremental modes are the two most common methods to practically implement the gradient-based training for such networks. Furthermore, since generalization is an important property and quality criterion of a trained network, pruning algorithms with the addition of regularization terms have been widely used as an efficient way to achieve good generalization. In this paper, we review the convergence property and other performance aspects of recently researched training approaches based on different penalization terms. In addition, we show the smoothing approximation tricks when the penalty term is non-differentiable at origin.

Key words: Gradient, feedforward neural networks, generalization, penalty, convergence, pruning algorithms.

1 Introduction

Multi-layer perceptron-type neural networks have been widely used in many real-life applications such as data analysis, trend analysis, classification, monitoring, control, clustering and pattern recognition. They automatically learn from observational data [1], [2]. Especially the so-called deep neural networks that have been used since since 2006 [3] have been recognized to achieve outstanding performance on many important problems in speech recognition, natural language processing, pattern recognition and computer vision [4].

The promising advantages of neural networks lie in their attractive and biologically inspired model. The models encompass data-driven learning techniques that adjust the weights without a specified function for the underlying model. We note that neural networks are mostly nonlinear models, which readily makes them flexible in dealing with real-world and complex tasks. More importantly, neural networks have been rigorously proved to be universal functional approximators, that is, they can approximate any given function with arbitrary accuracy [5].

As commonly known, the backpropagation (BP) algorithm is one of the mostly used techniques for training multi-layer neural networks [6], [7]. There are two main popular modes to implement the BP algorithm: batch mode and incremental mode [8]: For batch learning, the weights of networks are adjusted once and after an entire presentation of the training samples, which corresponds to the standard gradient method. The incremental learning, however, is a variation of the standard gradient method, which updates the weights once and after the presentation of each training sample.

The incremental training strategy may be sub-divided into three different modes in terms of the order that the samples are fed to the network. The first strategy is cyclic learning, whose order is randomly chosen in the initial step and then remains fixed during the whole training procedure. The second strategy is the almost-cyclic learning, where each sample is randomly chosen once per cycle. The last strategy is online learning which selects the samples in completely stochastic order in the whole training procedure.

The training process of a neural network may be stated as a "curve-fitting" problem. A network with "good" generalization means that the input-output mapping computed by the network performs well for testing data never used in training of the network. A network that is constructed to generalize well will generate a suitable input-output mapping. When, however, the training of a network concentrates on training samples, the network may memorize the training examples very well rather than fit the testing data. Such a phenomenon of "overfitting" usually indicates worse generalization.

There are three main factors on influencing the generalization of a trained network: the size of the training data, the network architecture, and the inherent complexity of the problem. How to determine the best architecture of network to achieve a good generalization becomes here an attractive aspect for studying the training properties.

Generally speaking, the optimal network architecture is one with the number of hidden units large enough to learn the examples and small enough to generalize well. To achieve a good generalization, the optimal network design depends on an appropriate tradeoff between reliability of training samples and goodness of the model. For BP learning, this tradeoff may be fulfilled by minimizing the total risk, expressed as a function of the weight vector $\mathbf{w}$ as follows

$$R(\mathbf{w}) = E_{av}(\mathbf{w}) + \lambda E_c(\mathbf{w}) \quad (1)$$

The first term is the normal BP error term, which depends on both the network model and the input samples. The second term represents the complexity of the network as a function of the weights, which is so-called penalty term, and λ is the penalization coefficient [9], [10]. The aim of (1) is to find a network with the simplest architecture possible and that is adequate to classify its input patterns.

Gradient-based method is a simple and popular learning algorithm for BPNN training. Some deterministic convergence results of both batch and incremental gradient algorithms for neural networks have been established in [11-19]. Boundedness of the weights during training turns out to be an important assumption to assure the convergent behavior. Interestingly, this prerequisite condition can be usually guaranteed by adding penalty terms. Due to different penalty terms, corresponding learning algorithms demonstrate different performance. In this paper, we will focus on the convergence analysis of BPNN with penalties and claim the distinction among them.

The rest of this paper is organized as follows. A brief description of batch and incremental (three different strategies) gradient-methods for BPNN training is given in Section 2. We describe the generally used penalties and pruning schemes in Section 3, while some specialized tricks for non-differential penalty functions will be stated in the Section 4. The asymptotic and deterministic convergence results are separately listed for different network modes in Section 5 and 6. In the last section, we offer a brief conclusion and some remarks.

2 Gradient method based algorithm for BPNN

Gradient descent method is a first-order optimization algorithm. It is often used to train BPNNs. To find a local minimum of error function, one takes steps proportional to the negative of the gradient of the error function in the weight space.

We consider a feedforward networks with three layers. Suppose that the training sample set is $\{\mathbf{x}^j, \mathbf{o}^j\}_{j=1}^J$, where $\mathbf{x}^j$ and $\mathbf{o}^j$ are the input and the desired output of the j-th sample.

For given input $\mathbf{x}^j$ $(j = 1, \cdots, J)$, the network output is denoted by $\mathbf{y}^j$. Correspondingly, the error produced at the output of the j-th input sample is defined by

$$e(\mathbf{w},\mathbf{x}^{j}) = \left\| \boldsymbol{o}^{j} - \boldsymbol{y}^{j} \right\| \tag{2}$$

where $\boldsymbol{y}^{j}$ is the corresponding actual output, $\mathbf{w}$ is the total weights. Summing the error-energy contributions of all the neurons in the output layer, the total instantaneous error energy of the whole networks can be expressed as follows

$$\boldsymbol{E}(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{J} e^{2}(\mathbf{w},\mathbf{x}^{j}) = \frac{1}{2}\sum_{j=0}^{J-1} \left\| \boldsymbol{o}^{j} - \boldsymbol{y}^{j} \right\|_{2}^{2}. \tag{3}$$

Naturally, the above error function is a function of all the adjustable weights of the multilayer networks. Depending on how the supervised learning of the BP neural networks is performed, two main different learning algorithms-namely, batch and incremental learning, as the following discussed in the context of gradient descent methods. We note that there are three specific incremental algorithms differ by the ordering of the training samples as online, cyclic and almost-cyclic learning [8].

2.1 Batch-mode learning

The popularity for the supervised training of multilayer neural networks has been enhanced by the development of the back-propagation algorithm. Gradient method is widely used to train the back-propagation algorithms.

In the bath-mode learning, adjustments to the weights of BPNN are performed after the presentation of all the J samples, or after an epoch. The cost function for batch learning is defined by the error function $\boldsymbol{E}(\mathbf{w})$, that is,

$$\boldsymbol{E}(\mathbf{w}) = \frac{1}{2}\sum_{j=0}^{J-1} e\left(\mathbf{w},\mathbf{x}^{j}\right) = \frac{1}{2}\sum_{j=0}^{J-1} \left\| \boldsymbol{o}^{j} - \boldsymbol{y}^{j} \right\|_{2}^{2} \tag{4}$$

Adjustments to the weights are carried on an epoch-by-epoch basis. The gradient of the error function is given by $\boldsymbol{\nabla E}_{\mathbf{w}}(\mathbf{w})$. Starting from an initial value $\mathbf{w}^{0}$, the weights $\{\mathbf{w}^{m}\}$ are interactively updated by

$$\mathbf{w}^{m+1} = \mathbf{w}^{m} - \eta \boldsymbol{E}_{\mathbf{w}}(\mathbf{w}^{m}),\ \ m = 0,1,2,\cdots. \tag{5}$$

We note that the batch-mode learning is completely deterministic and directly stems from the standard gradient method. It is clear to see that it requires large storage which is inconvenient in hardware applications.

2.2 Incremental learning

In the incremental method of supervised learning, adjustments to the weights of multilayer neural networks are performed on a sample-by-sample basis. That is, the weight updating takes place after each presentation of one drawn training sample. Due to the difference of ordering for the samples fed into the network, we distinguish three popular incremental learning strategies: cyclic learning, almost-cyclic learning and on-line learning.

2.2.1 Cyclic learning

Cyclic learning is a learning with a fixed cycle. Before training, the feeding order of training samples is randomly drawn and then fixed in the whole training procedure.

Given an initial weight $\mathbf{w}^0$, the cyclic learning updates the weights interactively by

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla e\left(\mathbf{w}^{mJ+j}, \mathbf{x}^j\right) \tag{6}$$

where m is the m-th iteration epoch, $\nabla e\left(\mathbf{w}^{mJ+j}, \mathbf{x}^j\right)$ is the gradient of the instantaneous error function with respect to the total weight. We note that the $(j+1)$-th weights updating of the m-th cycle depends on the j-th training sample.

2.2.2 Almost-cyclic learning

The training process of almost-cyclic learning consists of training cycles in which each of the samples is fed into the network exactly once. In addition, the order of sample presentation is continually drawn at random after each learning cycle.

For any given initial weight vector $\mathbf{w}^0$, the almost-cyclic learning changes the weights as follows

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla e\left(\mathbf{w}^{mJ+j}, \mathbf{x}^{m(j)}\right) \tag{7}$$

For the m-th training epoch, let $\left\{\mathbf{x}^{m(1)}, \mathbf{x}^{m(1)}, \cdots, \mathbf{x}^{m(J)}\right\}$ be a stochastic permutation of the $\{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^J\}$. This is the essential difference between cyclic and almost-cyclic learning.

2.2.3 On-line learning

For each online learning step, one of the training samples is randomly drawn from the training set and presented to the network. Online learning is a special incremental learning where the weight updating takes place after each presentation of randomly chosen training samples.

For any initial weight $\mathbf{w}^0$, the weights are iteratively updated by the following formula

$$\mathbf{w}^{m+1} = \mathbf{w}^m - \eta_m \nabla e\left(\mathbf{w}^m, \mathbf{x}^{r(m)}\right), \quad (8)$$

Where $m \in \mathbb{N}$ is the m-th iteration number, and r(m) is randomly chosen from $1,2,\cdots,J$. We make it clear that there is no training cycles for online learning. Each update of weight only depends on the randomly chosen sample.

3 Penalization and Network Pruning

To obtain a good generalization for a trained network, it is a popular strategy to add penalty terms to the standard error function. The penalties as below modify the objective function and the gradient based BP algorithm effectively prunes the network by pushing redundant weights to zero during training. Then the trained network performs as a smaller system with good generalization. The specific forms of these penalty functions are as follows:

$\phi(\mathbf{w}) = \sum_{w_{ij} \in \mathbf{w}} w_{ij}^2,$ (9) Weight Decay (L_2 Regularizer) [11-16, 20-22]

$\phi(\mathbf{w}) = \sum_{w_{ij} \in \mathbf{w}} \frac{(w_{ij}/w_0)^2}{1+(w_{ij}/w_0)^2},$ (10) Weight Elimination [23-26]

$\phi(\mathbf{w}) = \sum_{w_{ij} \in \mathbf{w}} |w_{ij}|^{\frac{1}{2}},$ (11) $L_{1/2}$ Regularizer [27], [28]

$\phi(\mathbf{w}) = \sum_{\mathrm{w}_i \in \mathbf{w}} \|\mathrm{w}_i\|,$ (12) Group Lasso penalty [29]

When a network is to be pruned, it is a common choice to add a penalty term with the sum of the squared weights. This quadratic penalty results in discouraging the weights from taking large values. However, this penalty term causes all weights to decay exponentially to zero at the same rate and disproportionately penalizes large weights.

To remedy this problem, the Weight Elimination penalty function has been proposed in [23, 24]. It penalizes small weights to decay at a higher rate than large weights by choosing suitable learning rate and penalization coefficients. However, a disadvantage of this penalty is that it can't distinguish between large and very large weights.

An $L_{1/2}$ regularizer was proposed in [27, 28] which is a nonconvex penalty. The $L_{1/2}$ regularizer is observed to have many promising properties such as unbiasedness, sparsity and Oracle properties. Particularly, the solution of the $L_{1/2}$ regularizer delivers better sparsity than that of the L_1 regularizer. Then, many references employ the $L_{1/2}$ regularizer of the weights as a penalty term. The experiments shoe that $L_{1/2}$ penalty forms better pruning achievement than weight Decay and Weight Elimination.

We note that the $L_{1/2}$ regularizer penalizes the weights individually, that is, some of the weights which connecting a neuron are decreased to zero while other weights are still retain with a large value. Thus, it may not prune the neurons at a group manner.

Group Lasso has been introduced in [29] as an extension of the so-called Lasso, which encourages sparsity at a group level. It is an intermediate penalty function between the L_1 penalty in Lasso and the L_2 penalty (weight Decay). Naturally, a novel penalty term has been investigated by borrowing the Group Lasso idea to train the BP neural networks.

It is clear to see that the penalty terms of the above function (11) and (12) are not differentiable at the origin. This may lead to difficulties on both theoretical analysis and numerical simulations, when the weights are very close to zero. It is a popular strategy to approximate the non-differential penalty term with smoothing functions.

For any finite dimensional vector $\boldsymbol{z}$, we introduce following smoothing functions.

1)"Sqrt Form"

$$h(\mathbf{z})=\begin{cases} \|\mathbf{z}\| & \|\mathbf{z}\| \geq \alpha, \\ \sqrt{\|\mathbf{z}\|+\alpha}+\alpha-\sqrt{\alpha^2+\alpha}, & \|\mathbf{z}\| < \alpha, \end{cases} \tag{13}$$

2)"Quadratic Form"

$$h(\mathbf{z})=\begin{cases} \|\mathbf{z}\|, & \|\mathbf{z}\| \geq \alpha, \\ \frac{\|\mathbf{z}\|^2}{2\alpha}+\frac{\alpha}{2}, & \|\mathbf{z}\| < \alpha, \end{cases} \tag{14}$$

3)"Quartic Form"

$$h(\mathbf{z})=\begin{cases} \|\mathbf{z}\|, & \|\mathbf{z}\| \geq \alpha, \\ -\frac{\|\mathbf{z}\|^4}{8\alpha^3}+\frac{3\|\mathbf{z}\|^2}{4\alpha}+\frac{3\alpha}{8}, & \|\mathbf{z}\| < \alpha, \end{cases} \tag{15}$$

4)"Sextic Form"

$$h(\mathbf{z})=\begin{cases} \|\mathbf{z}\|, & \|\mathbf{z}\| \geq \alpha, \\ \frac{\|\mathbf{z}\|^6-15\alpha^2\|\mathbf{z}\|^4+75\alpha^4\|\mathbf{z}\|^2+35\alpha^6}{96\alpha^5}, & \|\mathbf{z}\| < \alpha, \end{cases} \tag{16}$$

where the smoothing parameter $\alpha \leq 1$ is a fixed positive constant. We note that the approximations are much closer to the original absolute function for the higher orders of the smoothing functions.

4 Asymptotic Convergence Analysis

For online learning strategy, the theoretical results mainly perform asymptotic convergent behaviors since the training sequence is absolute randomly generated from the training data set. In [13], the convergence results of online BP neural networks with L_2 penalty were based on the following assumptions.

A1) Each training sample is randomly chosen from the training set $\{\mathbf{x}^{\boldsymbol{j}}, \mathbf{y}^{\boldsymbol{j}}\}_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{J}}$ with independent identical distribution;

A2) The activation functions $\boldsymbol{g}$ and $\boldsymbol{f}$ are twice continuously differentiable on $\mathbb{R}$. Moreover, $\boldsymbol{g}, \boldsymbol{f}, \boldsymbol{g}', \boldsymbol{f}', \boldsymbol{g}''$ and $\boldsymbol{f}''$ are uniformly bounded on $\mathbb{R}$.

A2') The activation function $\boldsymbol{g}$ is twice continuously differentiable on $\mathbb{R}$. Moreover, $\boldsymbol{g}, \boldsymbol{g}', \boldsymbol{g}''$ are uniformly bounded on $\mathbb{R}$. $\boldsymbol{f}(\boldsymbol{t}) = \boldsymbol{t}$ for all $\boldsymbol{t} \in \mathbb{R}$.

A3) $\{\boldsymbol{\eta_m}\}$ is a decreasing positive sequence such that a) $\sum_{\boldsymbol{m}=\mathbf{0}}^{\infty} \boldsymbol{\eta_m} = \infty$, b) $\lim_{m\to\infty} \sup(\boldsymbol{\eta_m^{-1}} - \boldsymbol{\eta_{m-1}^{-1}}) < \infty$, and c) $\sum_{\boldsymbol{m}=\mathbf{0}}^{\infty} \boldsymbol{\eta_m^{d}} < \infty$, for some $d > 1$.

Theorem 4.1. Suppose that the above assumptions A1), A3), and either A2) or A2') hold. Let $\{\mathbf{w}^{\boldsymbol{m}}\}_{m\geq 1}$ be a sequence of weight vectors iteratively generated by (online equation) with arbitrary initial value $\mathbf{w}^{\mathbf{0}}$. Then, $\mathbf{w}^{\boldsymbol{m}} \to \mathbf{w}^*$ with probability 1, where $\mathbf{w}^*$ is the optimal weight.

Fault-tolerant BP NNs have been proposed for over two decades, which inject weight noise during training procedure. However, until recently, only a few sources discuss its convergent behavior [21], [22].

Actually, they focus on two kinds of online fault-tolerant BP NNs: node fault and weight noise injections. The online node fault injection-based algorithm is that the hidden nodes randomly output zeros during training. For weight noise injections, there are mainly two types of weight noise injection-based algorithms during each step of training: multiplicative weight noise and additive weight noise injections with weight decay penalty (L_2 regularizer).

On the basis of the different fault strategies in training process, the corresponding objective functions are established, the boundedness of weight sequence and the asymptotic convergence results are rigorously proved. For brevity, we only list the convergence results for weight noise injection-based BPNN under mild conditions of activation function and learning rates [21].

A1) The activation function is set to be the common sigmoid function $f(x) = \frac{1}{e^{-x}}, x \in \mathbb{R}$.

A2) The learning rates $\boldsymbol{\eta_m}$ satisfy that $\boldsymbol{\eta_m \to 0}, \sum_{m=0}^{\infty} \boldsymbol{\eta_m} = \infty$ and $\sum_{m=0}^{\infty} \boldsymbol{\eta_m^2} < \infty$.

Theorem 4.2. Suppose that the activation function is with the sigmoid function in A1). The weight sequence $\{\mathbf{w}^{\boldsymbol{m}}\}_{m\geq 1}$ is iteratively generated by (online equation) with arbitrary initial value $\mathbf{w}^{\mathbf{0}}$. The penalization coefficient α is some positive constant. In addition, if the assumption A2) is also valid. Then, $\mathbf{lim}_{m\to\infty} \boldsymbol{\nabla} E(\mathbf{w}^{\boldsymbol{m}})\mathbf{=0}$ with probability 1, where $E(\mathbf{w}^{\boldsymbol{m}})$ is the established objective function.

5 Deterministic Convergence Analysis

5.1 Batch-mode learning with $L_{1/2}$ regularizer

The idea of $L_{1/2}$ regularizer has been successfully applied in variable selection and feature extraction problems in high dimensional data analysis. It is a nonconvex penalty and possesses many promising properties such as unbiasedness, sparsity and oracle property. Thus, it has been introduced into the batch gradient learning algorithm for the pruning BP NNs in [19].

Consider a single hidden-layer network consisting of p input neurons, q hidden neurons and 1 output neuron. Let $w_0 = (w_{10}, w_{20}, \cdots, w_{q0})^T \in \mathbb{R}^{\boldsymbol{q}}$ be the weight vector which connects the hidden nodes and the output node, and denote $w_i = (w_{i1}, w_{i2}, \cdots, w_{ip})^T \in \mathbb{R}^{\boldsymbol{p}}$ as the weight vector between the input nodes and the i-th hidden node. Let $\mathrm{g}: \mathbb{R} \to \mathbb{R}$ be the activation function of the hidden and output layers. Define a vector-value function $\mathrm{G}: \mathbb{R}^{\boldsymbol{q}} \to \mathbb{R}^{\boldsymbol{q}}, \mathrm{G}(\mathbf{x}) = (g(x_1), g(x_2), \cdots, g(x_q))^T \in \mathbb{R}^{\boldsymbol{q}}$, for $\mathbf{x} = (x_1, x_2, \cdots, x_q)^T \in \mathbb{R}^{\boldsymbol{q}}$. Suppose that $\{\mathbf{x}^{\boldsymbol{j}}, \boldsymbol{o}^{\boldsymbol{j}}\}_{\boldsymbol{j=1}}^{\boldsymbol{J}}$ are the given bounded training samples. The error function with the $L_{1/2}$ regularization penalty term is denoted by the following:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{J}(\mathbf{o}^j - g(\mathbf{w}_0 \cdot G(\mathbf{vx}^j)))^2 + \lambda \sum_{i=1}^{q}\sum_{k=0}^{p}|w_{ik}|^{\frac{1}{2}}, \tag{17}$$

We note that the $L_{1/2}$ regularization term in (11) is non-differentiable at the origin, which leads to more difficulties in theoretical analysis. More importantly, it is inevitable that the oscillation phenomenon will appear in numerical simulations. To overcome this drawback, a modified $L_{1/2}$ regularization term is presented by employing a smoothing function to approximate the absolution value function, which results in the new error function.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{J}(\mathbf{o}^j - g(\mathbf{w}_0 \cdot G(\mathbf{v}\mathbf{x}^j)))^2 + \lambda\sum_{i=1}^{q}\sum_{k=0}^{p}\sqrt{f(w_{ik})}, \tag{18}$$

where $f(x)$ is as the following piecewise polynomial function:

$$f(x)=\begin{cases}|x|, & if|x| \geq a,\\ -\frac{x^4}{8a^3}+\frac{3x^2}{4a}+\frac{3a}{8}, & if|x| < a,\end{cases} \tag{19}$$

Starting with an initial value $\mathbf{w}^0$, the weights $\{\mathbf{w}^n\}$ are iteratively updated by:

$$\mathbf{w}^{m+1}=\mathbf{w}^m\text{-}\eta\nabla E(\mathbf{w}^m), \tag{20}$$

where $\nabla E(\mathbf{w}^m)$ represents the gradient of $E(\mathbf{w}^m)$ with respect to $\mathbf{w}$, and the learning rate $\eta > 0$ is a constant.

To show the convergence results of the batch gradient method with smoothing $L_{1/2}$ regularization penalty, some sufficient conditions are as follows:

A1). $|g(t)|, |g'(t)|, |g''(t)|$ are uniformly bounded for $t \in \mathbb{R}$;

A2). $\|\mathbf{w}_0^m\|$ (m=0, 1, 2,⋯) are uniformly bounded;

A3). η and λ are chosen satisfy. $0 < \eta < \frac{1}{M\lambda + c}$, where $M = \frac{\sqrt{6}}{4\sqrt{a^3}}$ and c is a given positive constant;

A4). There exists a compact set Φ such that $\mathbf{w}^m \in \Phi$ and the set $\Phi_0 = \{\mathbf{w} \in \Phi : \nabla E(\mathbf{w}) = 0\}$ contains finite points.

Theorem 5.1. Let the error function be defined by (18), and the weight sequence $\{\mathbf{w}^m\}$ be generated by the iteration algorithm (20) for an arbitrary initial value. If assumption A1)-A3) are valid, then we have

(i). $E(\mathbf{w}^{m+1}) \leq E(\mathbf{w}^m)$, $m = 0, 1, 2, \cdots$;

(ii). There exists $E^* \geq 0$, such that $\lim_{n\to\infty} E(\mathbf{w}^m) = E^*$;

(iii). $\lim_{n\to\infty}\|\nabla E(\mathbf{w}^m)\| = 0$.

Furthermore, if assumption A4) also holds, then we have the following strong convergence;

(iv). There exists a point $\mathbf{w}^* \in \Phi_0$ such that $\lim_{n\to\infty} \mathbf{w}^m = \mathbf{w}^*$.

5.2 Cyclic learning with $L_{1/2}$ regularizer

Cyclic learning algorithm is one of the popular incremental algorithms compared with the batch mode training for BPNN. In addition, incremental learning algorithm is more efficient in term of both storage and computational burden. Based on the better pruning performance of $L_{1/2}$ regularizer, it is then presented as a penalty term for cyclic tearing of BP neural networks.

A modified smoothing $L_{1/2}$ regularizer has been proposed due to the non-differentiable ability at the origin. The smoothing error function is then defined for cyclic mode training procedure

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{o}^j - g(\mathbf{w}_0 \cdot G(\mathbf{v}\mathbf{x}^j)))^2 + \lambda \sum_{i=1}^{q} \sum_{k=0}^{p} \sqrt{f(w_{ik})}, \tag{21}$$

Where $f(x)$ is identical to the above definition (3), $j = 1, \cdots, J$. For any given initial weight $\mathbf{w}_0$, the weight sequence $\{\mathbf{w}^m\}$ is iteratively generated by:

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla E\left(\mathbf{w}^{mJ+j}\right), \tag{22}$$

where $m = 0, 1, 2, \cdots$; $j = 1, \cdots, J$; η_m is the learning rate in the m-th training epoch.

Let $\Phi_0 = \{\mathbf{w}: \nabla E(\mathbf{w}) = 0\}$ be the stationary point set of the error function $E(\mathbf{w})$. The following assumptions are imposed for the convergence of the cyclic BPNN with $L_{1/2}$ penalty term [18].

A1) $|g(t)|$ and $|g'(t)|$ are Lipschitz continuous for $t \in \mathbb{R}$;

A2) The learning rates η_m satisfy that

$$0 < \eta_m < 1 \text{ , and } \textstyle\sum_{\mathrm{m}=0}^{\infty} \eta_m < \infty.$$

Theorem 5.2. Let the error function $E(\mathbf{w})$ be defined by (21). $\mathbf{w}^0$ be an arbitrary initial value, and the weight sequence $\{\mathbf{w}^m\}$ be generated by the iteration algorithm (22). Assume the conditions A1) and A2) are valid, then there exists an unique $\mathbf{w}^* \in \Phi_0$, such that

$$\lim_{n\to\infty} \mathbf{w}^m = \mathbf{w}^*.$$

$$\lim_{m\to\infty} \|\nabla E(\mathbf{w}^m)\| = \|E(\mathbf{w}^*)\| = 0.$$

5.3 Almost cyclic learning with L_2 regularizer

For almost-cyclic learning with L_2 penalty, each training sample is chosen with a stochastic order and is fed exactly once in each training epoch. Let $\left\{\mathbf{x}^{m(1)}, \mathbf{x}^{m(2)}, \cdots, \mathbf{x}^{m(J)}\right\}$ be a stochastic permutation of the samples set $\{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^{J-1}\}$. The learning rate of training procedure is fixed as $\eta_m > 0$ in the m-th epoch. For fixed weight $\mathbf{w}$, the output error for the j-th iteration is defined as

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{o}^j - f(\mathbf{u} \cdot G(\mathbf{v}\mathbf{x}^{m(j)})))^2 + \lambda\|\mathbf{w}\|^2 \tag{23}$$

The weights are iteratively updated as following

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla E\left(\mathbf{w}^{mJ+j}, \mathbf{x}^{m(j)}\right), \tag{24}$$

where $\nabla E\left(\mathbf{w}^{mJ+j}, \mathbf{x}^{m(j)}\right)$ is the gradient with respect to $\mathbf{w}$ of the j-th iteration in the m-th epoch

Let $\Omega_0 = \{\mathbf{w}: \nabla E(\mathbf{w}) = 0\}$ be the stationary point set of the error function (23). Denote $\Omega_{0,s} \subset \mathbb{R}$ be the projection of Ω_0 onto the $s-th$ coordinate axis, that is,

$$\Omega_{0,s} = \left\{\mathbf{w}_s \in \mathbb{R}: \mathbf{w} = (w_1, w_2, \cdots, w_s, \cdots w_{n(p+1)}) \in \Omega_0\right\}$$

For $s = 1, 2, \cdots, n(p+1)$. To guarantee the convergence of the algorithm, the following assumptions are required [16]:

A1). $g'(t)$ and $f'(t)$ are Lipschitz continuous on $\mathbb{R}$;

A2). $\eta_m > 0$, $\sum_{\mathrm{m}=0}^{\infty} \eta_m = \infty$ and $\sum_{\mathrm{m}=0}^{\infty} {\eta_m}^2 < \infty$;

A3). $\Omega_{0,s}$ does not contain any interior point for every $s = 1, 2, \cdots, n(p+1)$.

Theorem 5.3. Assume that conditions A1) and A2) are valid. Then starting from an arbitrary initial weight $\mathbf{w}^0$, the learning sequence $\{\mathbf{w}^m\}$ generated by (24) is uniformly bounded, that is, there exists a positive constant $C > 0$ such that

$$\|\mathbf{w}^m\| < C, \quad m = 0, 1, 2, \cdots$$

and satisfies the following weak convergence

$$\lim_{n\to\infty} \|\nabla E(\mathbf{w}^m)\| = 0.$$

Moreover, if the assumption A3) is also valid, there holds the strong convergence. There exists an unique $\mathbf{w}^* \in \Phi_0$, such that

$$\lim_{n\to\infty} \mathbf{w}^m = \mathbf{w}^*.$$

6 Conclusions

Different penalty terms that are applied for different learning modes demonstrate various convergence results. For online training, only L_2 regularizer is imposed to train the fault-tolerant BPNNs. The weight sequence is uniformly bounded during training. In addition, the asymptotic convergence results are obtained due to the absolute randomly chosen for the training samples and the randomly weight noise.

For batch-mode learning, it is simple to get the convergence results since the standard gradient descent method are carried out in the training process.

Moreover, the learning rate can be selected as a small positive constant. It is a promising point that the $L_{1/2}$ regularizer as introduced during training improves the pruning ability and generalization.

For cyclic and almost cyclic learning, under assumptions deterministic convergence results have been obtained with L_2 and $L_{1/2}$ regularizer penalties under specific on the learning rates and the activation functions, respectively.

An observation can be made that a smaller network having similar approximation error for the training samples performs much better on generalization. Adding penalization terms to an objective function is an efficient way to prune the redundant hidden neuron. It is important to pay special attention to the convergence analysis which guarantees the convergent network from theoretical point of view. In addition, we note that pruning performance should be a promising additional aspect to study networks convergence behavior.

Acknowledgments

The authors wish to thank the anonymous reviewers for careful error proofing of the manuscript and many insightful comments and suggestions which greatly improved this work.

This project was supported in part by the National Natural Science Foundation of China (No. 61305075), the China Postdoctoral Science Foundation (No. 2012M520624), Natural Science Foundation of Shandong Province (No. ZR2013FQ004), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20130133120014) and the Fundamental Research Funds for the Central Universities (No. 15CX05053A, 15CX08011A).

References

1. Hagan M. T., Demuth H. B., Beale M. H., 1996, *Neural networks design*. Boston ; London: PWS.
2. Haykin S. S., 1999, *Neural networks : a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J. ; London: Prentice-Hall.
3. Hinton G. E.Salakhutdinov R. R., Jul 2006, *Reducing the dimensionality of data with neural networks*, Science, Vol. 313, No. 5786, pp. 504-507.
4. LeCun Y., Bengio Y., Hinton G., 05/28/ 2015, Deep learning, Nature, Vol. 521, No. 7553, pp. 436-444.
5. Sutskever I., Hinton G. E., Nov 2008, *Deep Narrow Sigmoid Belief Networks Are Universal Approximators*, Neural Computation, Vol. 20, No. 11, pp. 2629-2636.

6. Werbos P. J., 1974, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D., Harvard University, Cambridge, MA.
7. Rumelhart D. E., Hinton G. E., Williams R. J., Oct 9 1986, *Learning Representations by Back-Propagating Errors*, Nature, Vol. 323, No. 6088, pp. 533-536.
8. Nakama T., Dec 2009, *Theoretical analysis of batch and on-line training for gradient descent learning in neural networks*, Neurocomputing, Vol. 73, No. 1-3, pp. 151-159.
9. Reed R., 1993, *Pruning algorithms-a survey*, Neural Networks, IEEE Transactions on, Vol. 4, No. 5, pp. 740-747.
10. Bishop C. M., 1993, *Curvature-driven smoothing: a learning algorithm for feedforward networks*, Neural Networks, IEEE Transactions on, Vol. 4, No. 5, pp. 882-884.
11. Wu W., Shao H., Li Z., 2006, Convergence of batch BP algorithm with penalty for FNN training, in *Neural Information Processing*, pp. 562-569.
12. Zhang H., Wu W., Yao M., 2007, Boundedness of a batch gradient method with penalty for feedforward neural networks, in *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*, pp. 175-178.
13. Zhang H., Wu W., 2009, *Boundedness and convergence of online gradient method with penalty for linear output feedforward neural networks*, Neural Process Lett, Vol. 29, No. 3, pp. 205-212.
14. Zhang H., Wu W., Liu F., Yao M., 2009, *Boundedness and convergence of online gradient method with penalty for feedforward neural networks*, Neural Networks, IEEE Transactions on, Vol. 20, No. 6, pp. 1050-1054.
15. Shao H., Zheng G., 2011, *Boundedness and convergence of online gradient method with penalty and momentum*, Neurocomputing, Vol. 74, No. 5, pp. 765-770.
16. Wang J., Wu W., Zurada J. M., 2012, *Computational properties and convergence analysis of BPNN for cyclic and almost cyclic learning with penalty*, Neural Networks, Vol. 33, pp. 127-135.
17. Yu X., Chen Q., 2012, *Convergence of gradient method with penalty for Ridge Polynomial neural network*, Neurocomputing, Vol. 97, pp. 405-409.
18. Fan Q., Zurada J. M., Wu W., 2014, *Convergence of online gradient method for feedforward neural networks with smoothing L 1/2 regularization penalty*, Neurocomputing, Vol. 131, pp. 208-216.
19. Wu W., Fan Q., Zurada J. M., Wang J., Yang D., Liu Y., 2014, *Batch gradient method with smoothing L1/2 regularization for training of feedforward neural networks*, Neural Networks, Vol. 50, pp. 72-78.
20. Leung C. S., Tsoi A.-C., Chan L. W., 2001, *Two regularizers for recursive least squared algorithms in feedforward multilayered neural networks*, Neural Networks, IEEE Transactions on, Vol. 12, No. 6, pp. 1314-1332.
21. Sum J., Chi-Sing L., Ho K., 2012, *Convergence Analyses on On-Line Weight Noise Injection-Based Training Algorithms for MLPs*, Neural Networks and Learning Systems, IEEE Transactions on, Vol. 23, No. 11, pp. 1827-1840.

22. Sum J. P., Chi-Sing L., Ho K. I. J., 2012, *On-Line Node Fault Injection Training Algorithm for MLP Networks: Objective Function and Convergence Analysis*, Neural Networks and Learning Systems, IEEE Transactions on, Vol. 23, No. 2, pp. 211-222.
23. Weigend A. S., Rumelhart D. E., Huberman B., 1991, Generalization by weight-elimination applied to currency exchange rate prediction, in *Neural Networks, IJCNN 1991 International Joint Conference on*, Seattle, pp. 837-841.
24. Weigend A. S.Rumelhart D. E., 1992, *Generalization through minimal networks with application to forecasting*: Defense Technical Information Center.
25. Rakitianskaia A., Engelbrecht A., 2014, Weight regularization in particle swarm optimization neural network training, in *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, pp. 1-8.
26. Thomas P., Suhner M. C., 2015, *A new multilayer perceptron pruning algorithm for classification and regression applications*, Neural Process Lett, pp. 1-22.
27. Xu Z., Zhang H., Wang Y., Chang X., 2010, L(1/2) regularization, Science China-Information Sciences, Vol. 53, No. 6, pp. 1159-1165.
28. Xu Z., Chang X., Xu F., Zhang H., 2012, *L(1/2) Regularization: A Thresholding Representation Theory and a Fast Solver*, Neural Networks and Learning Systems, IEEE Transactions on, Vol. 23, No. 7, pp. 1013-1027.
29. Yuan M., Lin Y., 2006, *Model selection and estimation in regression with grouped variables,* Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 68, pp. 49–67.

Banking Services And Distribution Channels – Evolution And Prospects

Łukasz Murowaniecki

Department of Computer Science in Economics
University of Lodz, Łódź, Poland
lukaszm@uni.lodz.pl

Abstract

The article presents the process of evolution of banking services and banking distribution channels. It deals with the history and identifies main factors that affecting the change. In the following discussion, it presents phenomena that affect the future appearance of banking. These include changes in the economic and technological environment.

Key words: Banking service, distribution channel, IT in banking.

1 Introduction

Banking is an area of dynamic changes caused by many factors. Because of the role that banks play in the economy and, more broadly, in our lives, it is worthwhile to look at this phenomenon. Those changes apply to business model, the range of activities and also technological solutions.

Below, it is presented the author look at the process of evolution of banking services and bank distribution channels. In addition, the current trends and prospects in this area are indicated.

Such consideration has been provoked by many interesting and revolutionary statements presented in banking environment. To mention one example, the CEO of mBank, one of the biggest and most innovative bank in Poland, said that the bank branch, cash and signature - the pillars of traditional banking - at an accelerated pace no longer correspond to the rhythm of our day [1].

2 Bank products and services in distribution channels

We cannot find in the bibliography a clear and settled set of terms to describe the subject of the banking business. Such terms as ‘operation’, ‘product’ and ‘services’ are used interchangeably in banking context.

The Polish Banking Law defines the concept of banking operations [2]. There are distinguished: accepting cash deposits, lending, payment services. The concept of banking (financial) services appears in EU directives. A service is any activity, which the bank performs for the customer (e.g. current account management, granting a loan, money transfer, credit card payment execution). On the other hand, in the practitioners language, the bank offers products, which is consistent with the nomenclature of marketing.

From the information systems analyst point of view, I have adopted the following interpretation of terms. The bank sells products to customers, as a part of its activities. Products can be both material objects (e.g. a safety box) or immaterial (e.g. current account, credit card). As a part of product maintenance, the bank provides services to the client. The availability of services and rules for their implementation are described in the regulations of the product.

We can distinguish the following groups of banking services:

- deposit,
- credit,
- settlement,
- foreign transactions handling,
- investment,
- capital,
- information and consulting,
- other (e.g. Storage items).

Another important term for further analysis is a distribution channel (or ‘delivery channel’). It can be defined as a medium by which a financial institution provides services to customers. [3]

There are following distribution channels:

- sale in the bank branch,
- direct sales,
- indirect distribution (via email, dealers, franchise, etc.)
- remote channels; in this group we can distinguish:
    - phone channels (phone service, teleservice, hotline)
    - home banking
    - self-service devices (ATMs, Infomats, bank kiosks, etc.)
    - Internet channel,
    - mobile channels.

It should be remembered that various banking services can be done through various distribution channels. Typically, a service is not closely linked to one channel. In banking practice, we can observe the presence of a matrix relationship between services and distribution channels. It is a time varying matrix. Firstly, because both the list of banking products and services as well as the list of distribution channels expand. In addition, the availability of a particular service in a particular channel also changes. Typically, banks try to expand this availability.

3 The evolution of distribution channels

From the customer point of view, the evolution of distribution channels is the most influencing factor on the perception of banking activities. Modern distribution channels demonstrate the bank competitiveness. It is something what can be used as a bank trademark.

As an example could be an investment of mBank between years 2012 and 2013 into refreshing and extension their Internet and mobile distribution channels. It was valued at over 100 million PLN [4, 5]. This investment allowed the bank to provide their customers a radically altered Internet Banking channel and a new mobile application. This change, further strengthened by several international prizes [6] for its innovative interface and functionality of the banking system, has become a very powerful tool for promotion of the whole bank.

Another example is the strategy of Alior Bank. In 2012 it entered the market by promoting itself as the only fully online bank in Poland, investing in the project the amount of 50 million PLN [7]. A characteristic feature of the bank was supposed to be an Internet channel enabling access for the realization of all services.

Looking historically it must be admitted that channels of selling banking products have come a long way. Originally, a distribution of products and services was closely linked to a bank's headquarters location, a distance from the customer, banking hours. Products selling required a direct contact of a bank employee with the client. In order to maintain this way of operating, banks have heavily expanded its distribution network, which included bank branches and other non-bank service points. The need to reduce the physical distance between the institution and the customer has become the basis for the development of cooperative banks and other type of shadow banks and credit unions.

The first revolution in access to banking services took place with the development of telecommunication networks. This allowed the implementation of self-service distribution channels. The first was a telephone channel. First, only for information services, and then also for simple operations (e.g. a transfer order to the defined recipient). Subsequently, a call center was introduced for marketing purposes, handling complaints and more complex operations (e.g. card restriction, requests for products).

Another revolution brought the development of electronics, including computer technology. Electronic distribution channels has been developed. The whole implementation of banking services in these channels is called as electronic banking or just e-banking.

The evolution of electronic banking took place in several directions, simultaneously with the development of new electronic devices and technologies. The most important are:

- ATM - a stationary device the size of a man. Initially it took the form of information terminal, then the functionality has grown. Currently, we have to do with such devices as Cash Dispenser, ATM, cash deposit machine, cashless ATMs.
- Television – a device present in almost every households. Use it to realize banking services is referred to as banking television. At first, because of a one-way communication channel was limited for information role. Nowadays, TV can perform similar functions like a computer.
- POS - together with the use of plastic bank cards, and more recently mobile phones, is widely used in the retail sector as a channel for clearing services.
- Personal computer - a multi-functional device. It is used in several different ways. Initially it was a tool to run applications such as home banking, allowing to register financial operation off-line (later also on-line) and then interchange them with the bank by phone line or Internet. Currently, dominated by use of a computer as a tool to access bank websites.
- Mobile phones - can be used as an access device for several distribution channels (telephone, Internet, mobile). Its form and functionality begins to overlap with the PC. An important difference is that the mobile phone includes a personalized SIM card that gives additional possibilities, for example it can be used for user authentication.

In the context of electronic banking we can distinguish two trends (significant from the point of view of further consideration): internet banking and mobile banking. Their distinguishing feature is the technology of the service bank user interface. It seems to be a wrong way to attempt distinguishing between these distribution channels by the type of device used. This is due to the fact that differences between the personal computer (usually related to Internet banking) and mobile phone (assigned to mobile banking) are blurring.

Internet banking is the realization of banking services using an interface that is made in Web technologies. So these are the most common Internet banking services currently available through web browsers. This access channel has undergone a similar path of evolution to other solutions. Initially, only for information services, and then expand its functionality with basic active operations. And now it offers most of the services available in the physical facilities of banks. The main limitation appears to be the lack of widely available electronic signature mechanism.

Mobile banking use applications created in mobile technologies. Their user interfaces are typically dedicated to devices of relatively small size (a palm size). The range of services offered in this channel is still growing and becomes a direct competitor to Internet channels.

4 Evolution factors

Looking for the assessment of the future of banking services, we must consider factors contributing to their past development. It can be expected that changes are the result of many processes, often influencing each other. It is also the thesis I can formulate. The evolution of banking services and distribution channels is the result of social, organizational and technological factors. These impacts overlap and create pressure to change the practice and scope of banking activities.

As social factors I would mention:

- Enrichment of the population, resulting in an increased interest in financial services. In response, banks extend their offer and develop the branch network.
- Increase customer awareness, resulting from an increase in the level of education and experience in dealing with banks. For banks, this means greater competition in the market, because clients less trust in an opinion of the bank employee and more often compare various solutions. It also allows to offer more advanced services.
- The need to have analytical information, resulting in the emergence of services such as credit simulations, tools such as Personal Financial Manager (PFM), trend analysis of capital markets.
- The need for communication. One symptom of this is the fact that the cell phone is a modern device that we very often have close to us. Furthermore, a large part of the population is constantly on-line. Banks have a possibility to keep in touch with customers.

Organizational factors are:

- The end of the gold specie standard as a monetary system.
- Popularization of bills of exchange and promissory notes usage, what influence on the development of services related to bank money.
- A tendency to minimize cash transactions, giving an impact on offering a range of banking products and services. An example would be the popularization of plastic cards in the 60s of the twentieth century.
- The development of cooperation between institutional entities. It resulted in agreements in the local and global scale (e.g. SWIFT organization) and achieving standardization of processes. Examples are SWIFT standards for the exchange of data between financial institutions, SEPA for payment exchanges in the Euro zone or EMV for smart cards.

- Changes in legal systems. It can take a form of new regulations (like new prudential regulations) but also deregulation. Regardless of the direction of change, fundamentally affect the ability to offer certain services by financial institutions.

The last group of factors is associated with the technology development. As the most important we can identify:

- Technologies of communication for a long distance (telegraph), which resulted in the acceleration of settlement of transactions in relation to mail with a paper document.
- Electronics and increased functionality of electronic devices simultaneously with their miniaturization. This created a possibility to offer new distribution channels (electronic banking).
- Computer technologies, as a special kind of electronic revolution. The continuous increase in computer performance influences on banking information systems and thus allows providing more complex services. Further progress in this area leads to popularization of such concepts as Big-Data (possibility of data mining in large data sets) or in-memory processing.
- Internet. As a new communication media it allowed to offer a new distribution channel - internet banking. It also gave an opportunity to apply a new business model – a virtual bank and a virtual branch in the traditional bank. Internet accelerated an information exchange. In extreme cases leading to exchange data in real time (popular examples are: payment transactions or invoices exchange). The ability to maintain a continuous stable and secure Internet connection technology has allowed the development of cloud computing, which also can lead to a new bank business model.
- Biometric technologies. Currently used primarily in client authentication. However, we can assume that wider use is still to come.
- Mobile technologies allow keeping in constant communication with the bank regardless of where the customer is. That feature has opened a new distribution channel – mobile banking.
- Geolocation, partly related to mobile technologies. Allows to have an access to geographical location of the customer and in that way to customize services stronger.
- Short-range wireless communication technologies (such as RFID, Bluetooth, NFC). Although known for a long time, recently they are also used in banking. Contactless payments are an examples of such application. Another example can be a beacon, a small device becoming popular in recent years. In banking it can be used, among other things, for better offer personalization [8].

5 Changes in bank environment

The main change in the business environment is a strong increase of competitiveness. Primarily, it refers to the relationship between domestic banks and global banks entering a new market. Furthermore there are new players. These include other non-bank financial institutions such as credit unions, social lending portals, but also those in other sectors of the economy – insurance, , telecommunications, media and IT companies.

Particularly two phenomena are noteworthy. The first is a global technology corporations, like: Google, Amazon, Apple, Facebook, Alibaba. These Large Digital Players enter new service markets, previously controlled by banks. The second phenomenon are small business – technological startups. Both groups function as so called Big Bang Disruptors [9],[11]. They are characterized by: high efficiency, low cost, effective cooperation in creating content, innovative content, enhanced personalization, advanced technology solutions. The life cycle of products offered by these companies is very dynamic. It is characterized by rapid growth in popularity at the beginning, a short time of being on top and a sudden rapid decline of interest. A single occurrence of such company is not relevant for the environment, but in the global scale there are many of them. Their life cycles overlap. This allows customers to move from one innovation (currently on top) to the another (now at the beginning of rapid growth). As a result, it is a need for continuously monitoring and responding by large entities (including banks) to the market situation.

An example of innovation affecting the banking market can be the immediate payment market of in Poland. For a long time execution of credit transfer was the domain of banks and was managed by the National Clearing House (KIR SA). Most often, transfers were settled within a few hours. At some point in time, some companies (e.g. Przelewy24, PayU) entered that market as intermediary party. Based on the mechanism of real-time internal transfers existing in core banking system, they built a system allowing for immediate credit transfers between banks. The next step was done by Blue Media company, who proposed the new system (BlueCash) with implementation of immediate transfers, what became a direct competition to National Clearing House services. To prevent losing potential income, banks had to respond to the demand. Hence, KIR SA launched an Express ELIXIR system in 2012.

Another example may be a virtual currency (e.g. Bitcoin), by which the banking sector is motivated to develop their own systems of electronic money.

Growing competition on the market, transforms it into a customer market. The level of customer satisfaction becomes more important for banks in a short period of time than profit maximization. Customers' needs must be identified and answered quickly. A method of achieving this goal is a technological development.

IT development as a bank priority requires changes in bank organization. The role and responsibility of IT departments increases. Generally, a structure of employment is changing in favor of technology professionals. Bank employees responsible for contacts with customers also need to have a deeper IT knowledge.

Next aspect is a software engineering methodology. Habitually, banks applies traditional cascading methodologies, what results in production cycles lasting for a few months. It is too long in a new dynamic bank environment. The aim is to shorten these cycles, for example using agile methodologies.

Next observable trend is the blurring of boundaries between bank and other entities offer. On the one hand, banks are entering new markets like insurance, telecommunications or media. From the other side, representatives of other sectors extend their offer of banking products and services. Due to legal restrictions, it requires a cooperation between a bank and other companies or purchasing a bank and its banking license. There are few such examples on the Polish market. Among them alliances concluded between telecom T-Mobile and Alior Sync bank, between telecom Orange and mBank. Another example is the Plus Bank that is a part of the capital group consisting a telecom (Polkomtel) and a media group (Cyfrowy Polsat SA). One of the effect of such cooperation is product bundling, what means offering several products of different kinds in one combined offer. This strategy is advertised as win-win game for the bank and its associates. Cooperated parties can share their best practices, distribution networks and technologies.

One of the most famous attempts to gather all these trends in the banking sector is the concept of Bank 3.0 defined by Brett King [10]. He says about banks after the recent global financial crisis, operating in a dynamic competitive environment. As a consequence banks have to evolve organizational and technological. King asks the question 'Why banking is no longer somewhere you go but something you do' and in that way he defines a new perception of banks by customers and the future role of banks. In this concept, in order to survive, banks should:

- Reduce the role of traditional branches and concentrate on customer support. Instead of that virtual branches should be popularize.
- Cooperate with social media.
- Make banking systems more flexible and adapt them to work in a cloud environment.
- Bank staff need to be more familiar with high-tech solutions.

Another interesting concept is promoted by Accenture. It proposes a new bank business model called The Everyday Bank [9,12]. According to this vision, banks have to broaden their range of non-financial services and introduce new digital distribution channels. The Everyday Bank has a central position in customer everyday life and will act as an intermediary between the client and other entities in the digital economy. In addition, customer services should be carried out within a so-called ‘omnichannel’ environment. It makes an assumption that the choice of distribution channel should not affect the perception of the process by the customer. Moreover, customer service process can be performed in different channels alternately.

6. Technology trends

As previously mentioned, technological innovations overlap. Before one reaches its peak of popularity, another one starts to gain market share rapidly. To stay in business, banks have to adapt and respond quickly to new innovations.

Key technologies that can have the greatest impact on evolution of banking services and distribution channels are:

- Mobile technologies. They form the fastest growing distribution channel and allow to offer new services – e.g. mobile payments.
- Electronic money.
- Biometrics. It is a group of methods widely used for authentication, to confirm the identity (equivalent of Personal Identification Number) or identification (equivalent of login).
- Cloud computing.
- Big Data – analyzing huge data repositories to discover new useful knowledge. It is a technology that allows better understanding of the customer behavior, anticipating customer’s needs and offering better suited methods of communication.
- Behavioral Systems - examining customer behavior, including location data, transaction characteristics.
- Social media – can be used as a new channel of communication with customers but also as a new distribution channel. As an example a social payment can be mentioned. In this solution a social media portal become a place where we can send or receive a payment.
- Geomarketing - it the use of geographical parameters to carry out marketing campaigns. In banking practice it is used to decide on the location of the banking infrastructure (like bank branches, terminals, POS), to adjust marketing efforts to target groups residing in the area.

The use of new technologies results in a number of new services offered by banks. Below there are two examples of services that enrich the standard functionality of banking services. They are not critical from customers point of view, but they increase attractiveness of the offer..

The first example is a service named mRabaty offered by mBank. It uses a mechanism of geolocation of customer phone cell so is dedicated to be used in the mobile channel. In that solution, system identifies discounts available for the customer, classifies them on the basis of some preferences and current customer location. Finally all identified proposals are marked on the map displayed on the phone screen.

Another example is a tool for analyzing and managing of customer's own financial data. It can use data mining methods, for example to classify payments according to transaction features and assign them to a specific expense category and proper line item in the personal budget. A general name for such solution is Personal Financial Manager (abbreviated to PFM). Many banks offer such a functionality in their transactional systems. Moreover, there are also many non-banking sites on the market that specialize in this area (e.g. national: kontomierz.pl, ifin24.pl, budzetdomowy.pl, eportfel.com; global: yodlee.com). There are significant limitations in the development of PFM tools. First of all, it is difficult to extract data about the history of transactions from various banking systems and collect then in one system. There is no universally recognized standard for exchanging such data. Usually PFMs use a not very stable and secure technique called 'screen scraping' or inefficient manual input of data. Furthermore, a problem of recording cash transactions remains unresolved. Regardless of existing problems, this type of tool can be treated as a response to the need for analytical data by advanced online users.

7 Conclusions

The surroundings or conditions in which banks operate is constantly changing. From the bank point of view, it determines the need for continuous monitoring and responding customer expectations. This also creates a business opportunity for others financial market players. It is one of the areas in which Polish banks and the whole economy may prove to be innovative.

For bank customers it results in constant change of banks' distribution channels and services. In banks' assumption these changes should be better suited to customer needs. Is that realize indeed? It is a subjective matter. It is not hard to find opinions that financial offers have become unnecessarily complicated and new services, except attractive appearance, have little added value.

References

1. *In three years smartphones became a main distribution channel for banking services*, Deloitte, PRNews.pl, 2013-06-17 (in Polish).
2. Polish Banking Law, The Act of 29 August 1997 r., (OJ of 1997. No. 140, item. 939) (in Polish).

3. Paterson R., *Compendium of Banking Terms in Polish & English*, Fundacja Rozwoju Rachunkowości w Polsce, Warszawa 2002, p.168.
4. *New mBank for 100 million PLN*, www.biznes.newseria.pl, 2012-03-29 (in Polish).
5. *A new logo and transactional system of mBank. 100 million PLN on rebranding*, www.wirtualnemedia.pl, 2013-06-05 (in Polish).
6. *New mBank is a most groundbreaking innovation in the world*, www.mbank.pl, 2013-10-12 (in Polish).
7. Samcik M., *Alior Sync ready to go live! But is this really the bank hit, we've been waiting for?*, samcik.blox.pl, 2012-06-13 (in Polish).
8. *What Beacons could do for banks and retailers?*, www.monitise.com, 2014-05-20.
9. Mazurek K., *The Everyday Bank*, Accenture 2014, www.alebank.pl (in Polish).
10. King Y. B., *Bank 3.0: Why Banking Is No Longer Somewhere You Go But Something You Do*, Wiley 2012.
11. Downes L., Nunes P., *Big-Bang Disruption*, Harvard Business Review, March 2013, hbr.org (2015-11-15).
12. *The Everyday Bank, A New Vision for the Digita Age*, Accenture 2015, www.accenture.com (2015-11-15).

PERFORMANCE ANALYSIS OF VIRTUAL COMPUTER NETWORK BASED ON CISCO CLOUD SERVICES ROUTER 1000V IN A PRIVATE CLOUD ENVIRONMENT

Stanislaw Lota[1], Marcin Markowski[2]

[1] Maria Sklodowska-Curie Upper-Secondary School Complex in Olesnica
Wojska Polskiego 67/69, 56-400 Olesnica, Poland
stanislaw.lota@gmail.com

[2] Department of Systems and Computer Networks,
Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
marcin.markowski@pwr.edu.pl

Abstract

Virtualization of physical network devices is a relatively new technology, that allows to improve the network organization and gives new possibilities for Software Defined Networking (SDN). Network virtualization is also commonly used for testing and debugging environments, before implementing new designs in production networks. Important aspect of network virtualization is selecting virtual platform and technology, that offer maximal performance with minimal physical resource utilization. This article presents a comparative analysis of performance of the virtual network created by the virtual CSR1000v and virtual machines running Windows 8.1 on two different virtual private cloud platforms: VMware vSphere 5.5 and Microsoft Hyper-V Server 2012 R2. In such prepared testbed we study the response time (delay) and throughput of virtual network devices.

Key words: virtual network, virtualization platforms, network performance,
Cisco Cloud Service Router 1000v, Hyper-V, VMware ESXi,

1 Introduction

Currently, the physical network devices such as routers, switches, and servers are the base of computer networks. Popularity of cloud computing forces the research for new network solutions, integrating physical network environments with virtual environments. For a few years works on transferring the functionality of physical network devices to their virtual counterparts and Software Defined Networking (SDN) are being conducted.

Network virtualization technology enables reproduction of the physical network using the virtual network devices running on a single physical server, offering the same functionality as the physical devices. Multiple virtual network devices, such as routers, switches, firewalls, intrusion detection systems, may be used in a virtualized environment. It gives the opportunity to organize network in better, more efficient manner [9]. Other advantages are the optimization of costs, short implementation time, faster network integration, reduction of cabling, independence from expensive physical devices, simpler and faster disaster or failure recovery. Virtualization ensures also high energy-efficiency, since many virtual systems (network devices or servers) utilize single hardware platform. Currently, the virtual computer network can be associated with existing network hardware providing full communication.

One of the important challenge, faced by network virtualization solutions is the performance deficiency. The operational delay, usually very small for hardware network devices, in virtualized devices is being often multiplied as the reason of underlying hypervisor software and hardware platform not optimized for network traffic processing. Also the maximal throughput, offered by virtual network solutions often becomes a bottleneck.

In this context, very important decision in virtual network designing and implementing process in the choice of optimal, efficient virtualization platform, that will not significantly decrease the network performance (Quality of Service - i.e. delay in the network; throughput, etc.). Then, we provide short study on the efficiency of two of the main virtualization solutions.

The main purpose of this article is to present and compare the performance of two virtualization platforms, VMware vSphere 5.5 and Microsoft Hyper-V Server 2012 R2, used as a base for virtual network solution. Experimental virtual network is built of Cisco Cloud Service 1000V Routers and virtual hosts. Performance has been studied in the area of Quality of Service of the virtual network i.e. bandwidth, delay and jitter. Chosen criteria will accurately determine whether network bandwidth corresponds to the capabilities offered by the devices, whether the traffic is not blocked on specific ports, or whether large packet loss and unstable operation of the network may be observed.

In section 2 we present short review of virtual network devices, section 3 gives the look at actual state of art in network virtualization studies. Then, in section 4 we formulate the research problem and present the testbed. Results of experiment are reported in section 5. Finally, we conclude the research in section 6.

2 Virtual Network Devices

The first implementation of virtual network device on PC platform was a computer equipped with two network adapters and specialized software. Such solutions were usually built on FreeBSD distributions and could act as a router, firewall, DHCP server, DNS, or VPN. Nowadays many project, such as pfSense, m0n0wall or VyOS, develop software packages that provides features similar to commercial hardware boxes, and often gaining additional functionalities and greater control of security [2]. These systems can be successfully installed in a VMware environment, as well as Hyper-V virtual machines. In more complex virtual environments, e.g. when we need to connect few virtual servers working on single physical server, we may use physical network device (router or switch) connected to the server to provide connection (and routing) between virtual machines [2]. Another, often chipper and less complex solution is to use virtual network device, located on the same physical server. We can now virtualize routers, switches, firewalls, intrusion detection and intrusion prevention systems (IDS/IPS), as well as load balancers, NetFlow collectors and less common network devices on one server with hypervisor system. These devices can work independently using the same hardware platform. Among virtual device solutions we may distinguish layer 2 and layer 3 switches, routers, firewall and others. An example of virtual Layer 2 switch are the Cisco Nexus 1000V (VMware Distributed Virtual Switch) and open source Open vSwitch. Features of Layer 3 switches are implemented in virtual switch Nexus 1000V, developed by Cisco. The Cisco Virtual Nexus1000V working under the control of the NX-OS and allows to create PVLAN, virtualized DMZ zones and implementation of the policies for advanced network security e.g. ACL together with QoS [4]. ASA 1000V Cloud Firewall is a virtualized version of a hardware firewall ASA 5500 series and provides protection coastline and Virtual Security Gateway (VSG) responsible for the protection of the network using VMware vShield APIs for internal security. For application of routing between virtual machines in a private, public or hybrid cloud environment, Cisco created a virtual router enabling selected functions of the operating system IOS-EX - Cisco Cloud Service Router 1000V. The virtual device is designed for deployment in data centers in the cloud and run as a virtual machine on servers that use virtualization platforms VMware ESXi, Citrix XenServer, Kernel Virtual Machine (KVM) and Microsoft Hyper-V [5]. Noteworthy is also the possibility of implementing Cisco Cloud Service Router 1000V on public cloud Amazon AWS and Microsoft Azure [6]. The main use of the virtual router is acting as a gateway to the WAN for multitenant and secure connection between the provider of public cloud and enterprise. These functions can be implemented using IPSec VPN (DMVPN, EasyVPN and FlexVPN), or MPLS (Multi-Protocol Label Switching). Cisco Cloud Service

Router 1000V is licensed base on a combination of performance and feature set. Virtual machine with CSR1000V deployed on servers requires from 1 to 4 virtual CPUs, from 2.5 to 4GB of RAM, depending on the performance and feature set, 8GB of disk space and three or more virtual NICs vNICs. The device can provide routing functions between virtual machines using protocols such as OSPF, EIGRP, and BGP, Multicast, LISP, GRE [6]. Another solution is to implement a cloud computing environment with virtual router Brocade Vyatta vRouter and community version VyOS. Brocade provides routing based on BGP Multipath, PBR, OSPF, Multicast technologies, IPsec VPN environments for physical, virtual and cloud-based environments. Vyatta vRouter can also function as a firewall but does not support MPLS [1].

3 Related Works

Virtual network operation seems to be important practical problem, but literature connected with the topic is very limited. The most interesting publication is [8], where authors present optimization problem where the objective is to minimize the network resource consumption with virtualization support (NFV-capable nodes), such that the service requirement (order of service chain traversal) for all the traffic flows is satisfied. In [10] authors propose a virtual network architecture for cloud computing and present research about virtual network which can provide communications for virtual resources in cloud computing. It can potentially reduce the global CO2 emission. Furthermore, without purchasing, operating, maintaining, and periodically upgrading local computing infrastructures, cloud computing can lower the cost of IT services for an enterprise. Multiple virtual networks can run simultaneously over a single physical infrastructure without interfering with each other. In this research virtualized network components such as links, bridges and routers were considered. This virtual network can provide the communication between virtual hosts with flexibility. Furthermore, the virtual network can run the customized routing protocol [10]. Furthermore, Ka Ching Chan and Mary Martin in [3] present an infrastructure enabling lecturers to design and set up experiments in not only traditional networking topics such as RIP, OSPF, BGP, and VLAN using a combination of physical and virtual networking devices, but also in the latest technologies such as server virtualization and network virtualization. They present the development of an integrated virtual and physical network infrastructure for the Internetworking Laboratory at La Trobe University's Bendigo campus. The infrastructure was setup with physical equipment including Cisco routers, Cisco switches, a number of Ubuntu Linux workstations, and a VMware ESXi server hosting a number of virtual machines including Vyatta routers, and

Ubuntu virtual desktops [3]. In article [11] authors evaluated performance of virtual router platforms based on Linux namespaces and show that hardware assisted virtual routers can achieve better aggregate throughput than a non-virtualized router on a multi-core platform [11]. It is noteworthy that in 2014 years Cisco in a white paper [7] describes technology Virtual Extensible LAN (VXLAN) and how to use CSR 1000V to route between VXLAN segments (VXLAN Layer 3 routing) in addition to switch Cisco Nexus 1000V support for VXLAN.

4 Research Problem and Experimental Environment

The main research problem considered in this paper is the evaluation performance of the network created by the virtual machines and the Cisco Cloud Services Router 1000V. Evaluation criteria are bandwidth, delay, and jitter. Bandwidth is the amount of data that can be sent over the network between two of its points, e.g. router – router or computer - computer in a unit of time. Bandwidth is measured in bits per seconds and is particularly important in the case of transferring large amounts of data over a network. The delay (latency) is the time needed for packet to flow between two designated points. The lower the latency, the better network performance, because the local infrastructure should not exceed a few milliseconds. Jitter is important for real-time transmissions, usually using UDP protocol, e.g. VoIP and multimedia transmissions.

Experimental environment was built on server equipped with 8 core FX-8150 processor clocked at reference period of 3.6 GHz, 16 GB of RAM and two 500GB disks HDD speed 7200rpm and 16MB cache.
Two experimental virtual cloud platforms were implemented on the physical server:

- VMware vSphere 5.5 ESXi 5.5 hypervisor virtualization platform. To create a private cloud environment uses Vmware vCloud Director, vCenter 5.5 vCenter SSO (Single Sign-On) and vSphere Web Client.
- Microsoft Hyper-V Server 2012 R2. To create a private cloud environment uses Microsoft System Center 2012 R2 Virtual Machine Manager 2012 R2.

In order to conduct performance measurement, virtual network based on the Cisco Cloud Service Router 1000V was implemented in both virtual environments. Topology of the virtual network is presented in the Fig. 1. The experimental virtual network consists of three virtualized Cisco Cloud Services Router 1000V, which were connected together by means of virtual interfaces based on virtual standard switches, to define between them three subnetworks. Each of the routers also handle a single network which contains virtual machines with Windows 8.1 Professional. Static routing was implemented in the virtual network. All experiments were performed in a client – server model. The first virtual machine with Windows 8.1 acted as the server, the other virtual machines Windows 8.1 acted as a client.

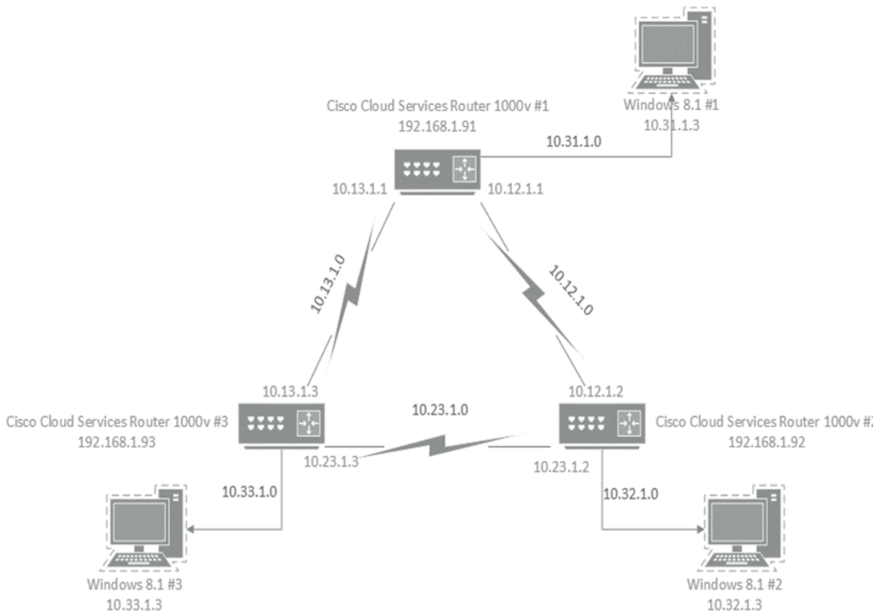

Figure 1. Virtual network topology

A tool Psping included in the PsTools package, which being extended version of the traditional tools Ping, was use for demand generating in the virtual network. The tool allows to check the availability of devices through ICMP, send test packets to any TCP port, perform the measurement of the delay and bandwidth in client - server architecture. Iperf tool was also used to determine the bandwidth between two computers, delay, loss of datagrams, delay variation. Iperf allows testing TCP and UDP on selected ports.

5 Results

The goal of first test was to examine the estimated packet round-trip time [in milliseconds] for 32-byte packet size. We performed 10 experiments, 1000 ping request in each. For each experiment we calculated minimal, maximal and average delay. All attempts were successful and no packets were lost. A large difference between the minimum and maximum delay for both of the test platform was observed during this experiment. Minimal response times for each experiment are presented in the Fig. 2. Lowest delay was noted for

VMware environments, while in the case of Hyper-V delay of each trial decreased in consecutive experiments. Lowest delay obtained in all experiments was 0.6ms for Hyper-V and 0.39 ms for VMware.

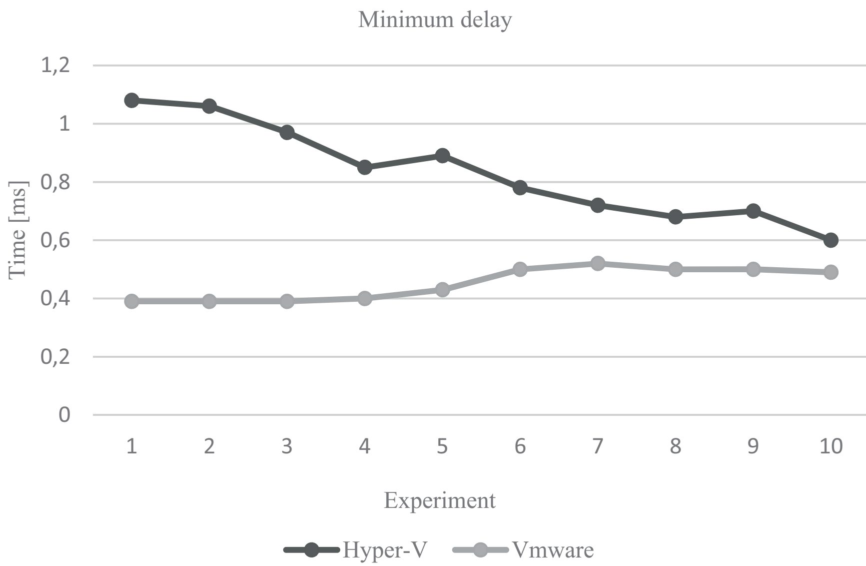

Figure 2. Minimal delay times in each experiment for 32B packet size

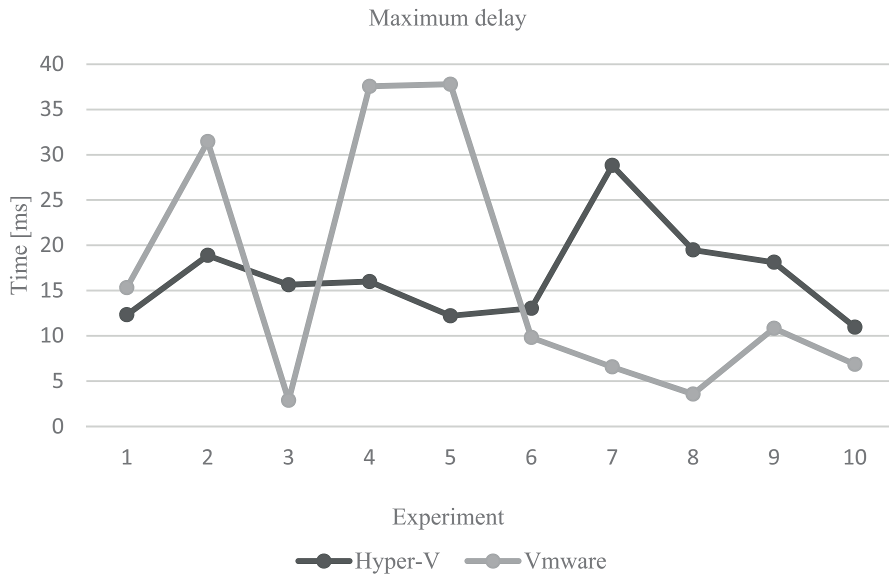

Figure 3. Maximal delay times for 32B packet size

Maximal response times for the same scenarios are presented in the Fig. 3, and average ones in the Fig. 4. VMware several times noted higher delay in comparison to the Hyper-V (in second, fourth and fifth experiment). The biggest delay fluctuated between 2.87 ms and 37.79 ms for VMware and from 11 to 28.91 ms in case of Hyper-V. Average time fluctuated around 2.50ms for Hyper-V, whereas for VMware within 0.60 ms.

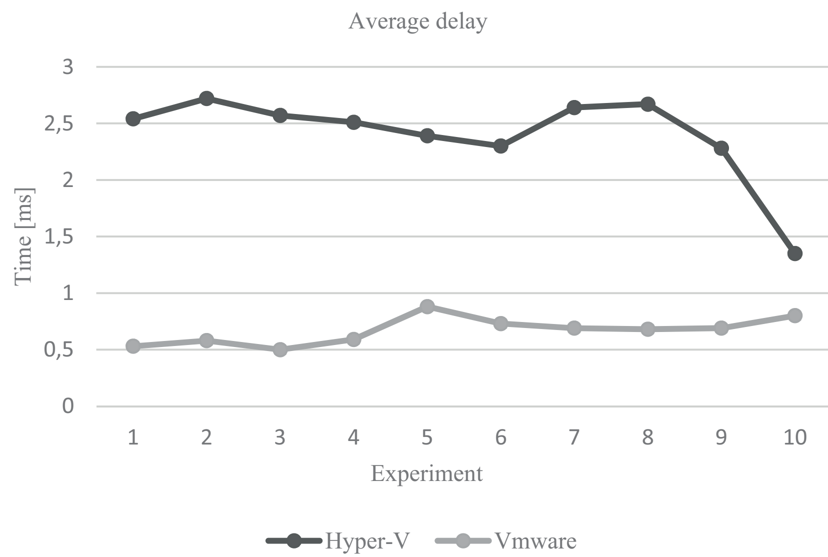

Figure 4. Average delay times for 32B packet size

The second test was to estimate the delay for 8192B packet size. As previous we performed 10 experiments, 1000 ping request each. Results of experiments are presented in the Fig. 5-7.

As we may observe from the results, for VMware environment delay is much more unpredictable and varies in range from 1 to 4000 ms, with the average delay around 300ms. Remarkably better results were obtained for Hyper-V platform, delay fluctuated in range from 7 ms to 38 ms with average around 28 ms.

Although average delay was considerably lower for Hyper-V, the lowest delays were observed for VMware–based virtualization platform. Compiling those results with results for previous experiments for 32B packets size, we may conclude that for some moments VMware is able to ensure better answer time, but generally is very unstable. Taking into account delay experiments, Hyper-V virtualization platform offers better and more stable efficiency.

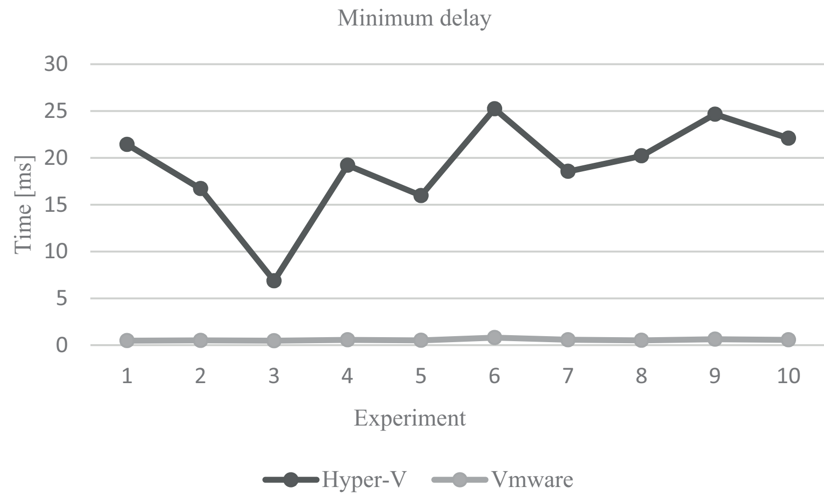

Figure 5. Minimal delay times for 8kB packet size

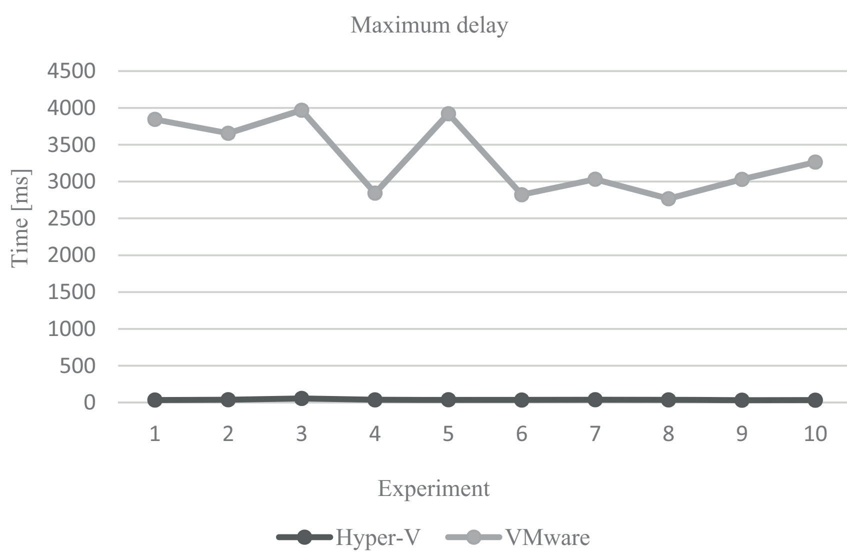

Figure 6. Maximal delay times for 8kB packet size

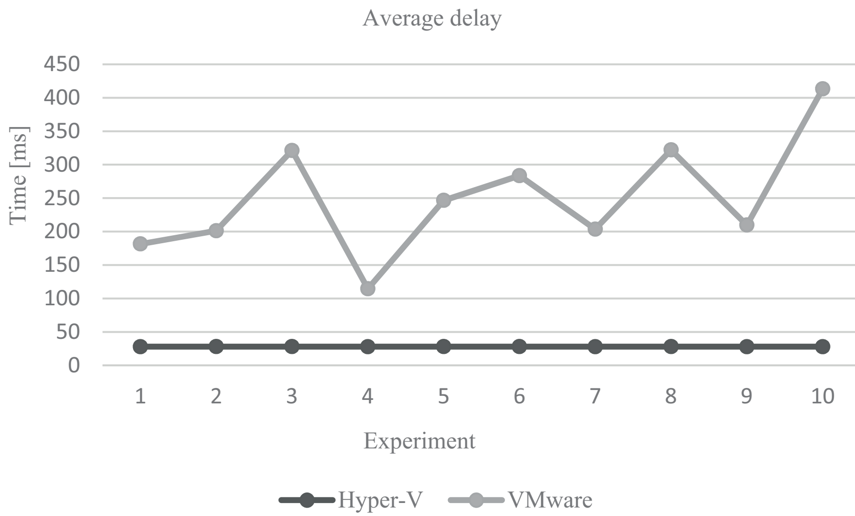

Figure 7. Average delay times for 8kB packet size

The aim of the third experiment was to estimate the capacity of the virtual network. As previously 10 experiments were performed for both evaluated platforms. In each experiment 10,000 packets were sent in the virtual network. Packets were generated with PsPing, and the size of packets was set to 8192 bytes. In the distribution for non-commercial use, Cloud Services Router 1000v offers a maximum bandwidth of 2.5 Mb/s, then we do not expect higher capacity in virtual network. Results of experiment – higher, lower and average throughput are presented in the Fig. 8-10. Unlike during previous experiments, VMware environment was more stable and generally offered better and regular traffic ability performance. Minimal throughput for VMware was newer below 250 Kb/s. In Hyper-V, the smallest throughput was 74.76 Kb / s, and fluctuated up to 236.23 Kb/s.

The greatest maximum throughput was reached in a Hyper-V environment. A little worse efficiency was offered by VMware. In the case of an average throughput results are very similar in both environments with a slight predominance of the VMware environment. Highest average throughput for VMware was 289.33 Kb/s and for Hyper-V 287.42 Kb/s. In the case of 8th attempt for Hyper-V we have noticed the lowest score of at least 331.27 Kb/s and average equal to 260.88 Kb/s.

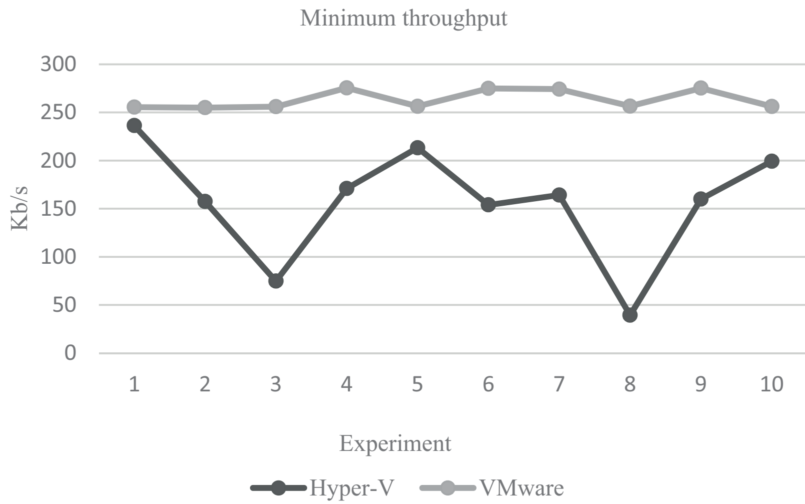

Figure 8. Minimal throughput in virtual networks

In the fourth test we have estimated bandwidth in virtual network using iperf traffic generator. The goal was to estimate the throughput in both directions (client-server and server-client). Results of experiments are presented in the Table 1. As compared to throughput of a server and a client server to a client in both environments have a difference in the throughput of 0.05 Mb/s.

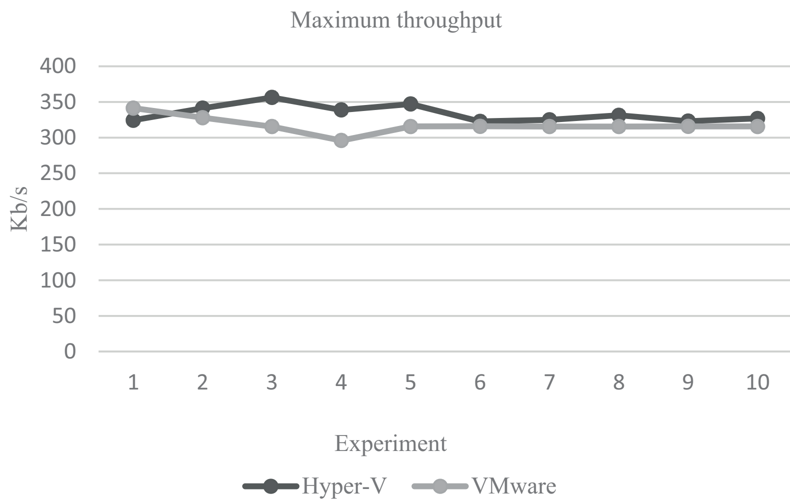

Figure 9. Maximal throughput in virtual networks

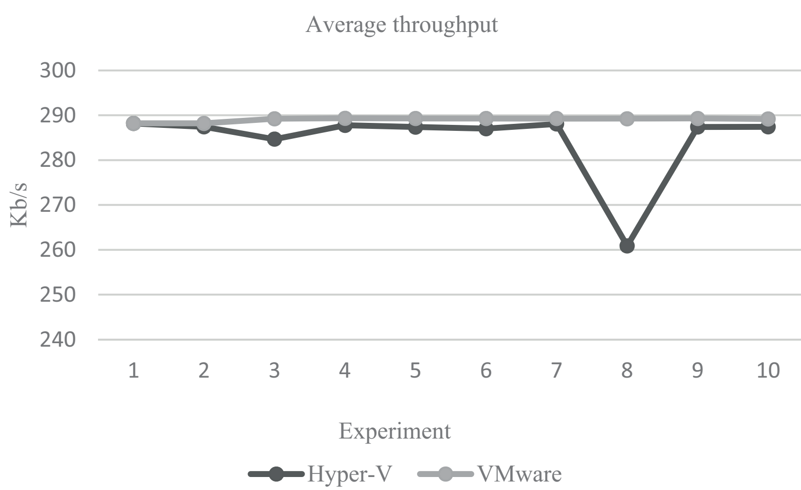

Figure 10. Average throughput in virtual networks

The next experiment consisted of assessing jitter and provide information about the number of packets lost by iperf. For Hyper-V smallest jitter was 0.130 ms, while the largest 0.5 ms. For VMware smallest jitter was 7.429 ms, and the largest 8.802 ms. In all of the experiments packet loss does not exceed 1%. Results of jitter studies are presented in the Table 2.

Table 1. Comparison of throughput for one and two-directions scenarios

Scenario	Hyper-V		Vmware	
	Min	Max	Min	Max
throughput by iperf	2.42 Mb/s	2.43 Mb/s	2.42 Mb/s	2.43 Mb/s
	Min	Max	Min	Max
throughput in both directions by iperf [client – server]	2.37 Mb/s	2.43 Mb/s	2.42 Mb/s	2.43 Mb/s
	Min	Max	Min	Max
throughput in both directions by iperf [server – client]	2.28 Mb/s	2.42 Mb/s	2.37 Mb/s	2.38 Mb/s

Table 2. Results of jitter studies

Experiment	Hyper-V	VMware
1.	0.5 ms	8.29 ms
2.	0.13 ms	8.802 ms
3.	0.397 ms	7.429 ms
4.	0.172 ms	8.248 ms
5.	0.194 ms	8.042 ms

6 Conclusion

Analyzing results of all performed experiments, we may conclude that Hyper-V virtualization platform offers better efficiency and better Quality of Service for virtual computer network. Most important conclusion is, that Hyper-V based virtual network behaves more stable and predictable with much lower delay and throughput fluctuation. We may recommend this platform rather than VMware, for virtualization of small virtual computer networks, built on one physical machine.

Quite important problem observed during experiments was the weak recurrence of the obtained results in both environments. Another problem was physical server performance, which is very well illustrated in performance tests for the delay and throughput. Probably, the source of those problems was too great load on the physical server processor, during which the hypervisor ESXi had to decide which virtual machine first should use the computational power of the processor. This leads to a situation where all traffic is handled by only one logical processor, while the other ones are asleep. VMware and Microsoft also points out, that the results of performance tests may be disrupted by the power plan of virtual machines running Windows 8.1. By default, Windows sets balanced mode of performance and energy consumption. Despite the efforts to define stable and identical test conditions, it turned out that the individual measurements for the delay and bandwidth can be quite significantly different in each of the trials. Moreover, we should observe the directions of development of these technologies, because similar situation may take place in the future, as in the case of cloud computing, which was a novelty two years ago, and today is an essential tool for facilitating daily life.

References

1. Brocade Communication Systems, 2016, *Brocade 5400 vRouter (formerly Vyatta 5400 vRouter) Data Sheet*, Brocade Communication Systems, Inc., San Jose, CA.
2. Ka Ching C., 2015, *Integration of Physical Equipment and Simulators for On-campus and Online Delivery of Practical Networking Labs*, Technical Report CSIT, La Trobe University, Bendigo, Australia.
3. Ka Ching C., Martin M., 2012, *An Integrated Virtual and Physical Network Infrastructure for a Networking Laboratory*, The 7th International Conference on Computer Science & Education.
4. Cisco Systems, 2016, *Cisco Nexus 1000V Switch for VMware vSphere. Cisco Systems*, Inc., San Jose, CA.
5. Cisco Systems, 2016, *Cisco Cloud Services Router 1000V 3.16 Series Data Sheet*, Cisco Systems, Inc., San Jose, CA.
6. Cisco Systems, 2016, *Q&A Cisco Cloud Services Router 1000V with Cisco IOS-XE® Software Release 3.10.*, Cisco Systems, Inc., San Jose, CA.
7. Cisco, 2016, *White Paper Flexible Workload Mobility and Server Placement with VXLAN Routing on Cisco CSR 1000V and Cisco Nexus 1000V.*

8. Gupta A., Farhan M., Chowdhury P., Tornatore M, Mukherjee B., 2015, *Joint Virtual Network Function Placement and Routing of Traffic in Operator Networks*, Department of Computer Science, University of California Davis.
9. Khan A., Zugenmaier A., Jurca D., Kellerer W., 2012, *Network Virtualization: A Hypervisor For the internet?,* IEEE Communications Magazine, January 2012 , pp 136-143.
10. Ruay-Shiung C. Chia-Ming W., 2010, *Green Virtual Networks for Cloud Computing*, Dept. of Computer Science and Information Engineering National Dong Hwa University Hualien, Taiwan, R. O. C.
11. Siraj M. R., Hidell M., Sjödin P., 2012, *PC-based Router Virtualization with Hardware Support*, 26th IEEE International Conference on Advanced Information Networking and Applications.

Information for Authors

Authors are invited to submit original papers, within the scope of the JACSM, with the understanding that their contents are unpublished and are not being actively under consideration for publication elsewhere.

For any previously published and copyrighted material, a special permission from the copyright owner is required. This concerns, for instance, figures or tables for which copyright exists. In such a case, it is necessary to mention by the author(s), in the paper, that this material is reprinted with the permission.

Manuscripts, in English, with an abstract and the key words, are to be submitted to the Editorial Office in electronic version (both: source and pdf formats) via e-mail. A decision to accept/revise/reject the manuscript will be sent to the Author along with the recommendations made by at least two referees. Suitability for publications will be assessed on the basis of the relevance of the paper contents, its originality, technical quality, accuracy and language correctness.

Upon acceptance, Authors will transfer copyright of the paper to the publisher, i.e. the Academy of Management, in Lodz, Poland, by sending the paper to the JACSM Editorial Office.

The papers accepted for publication in the JACSM must be typeset by their Authors, according to the requirements concerning final version of the manuscripts. However, minor corrections may have been carried out by the publisher, if necessary.

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm.

The first page of each paper should contain its title, the name of the author(s) with the affiliation(s) and e-mail address(es), and then the abstract and keywords, as show above. Capital letters must be applied in the title of a paper, and 14 pt. boldface font should be used, as in the example at the first page. Use: 11-point type for the name(s) of the author(s), 10-point type for the address(es) and the title of the abstract, 9-points type for the abstract and the key words.

For the main text, please use 11-point type and single-line spacing. We recommend using Times New Roman (TNR) fonts in its normal format. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided. With these sizes, the interline distance should be set so that some 43 lines occur on a full-text page.

The papers should show no printed page numbers; these are allocated by the Editorial Office.

Authors have the right to post their Academy of Management-copyrighted material from JACSM on their own servers without permission, provided that the server displays a prominent notice alerting readers to their obligations with respect to copyrighted material. Posted work has to be the final version as printed, include the copyright notice and all necessary citation details (vol, pp, no, …).

An example of an acceptable notice is:
“This material is presented to ensure timely dissemination of scholarly work, and its personal or educational use is permitted. Copyright and all rights therein are retained by the Copyright holder. Reprinting or re-using it in any form requires permission of the Copyright holder.

The submission format for the final version of the manuscript
and the MS Word template are available on our web site at:
http://acsm.spoleczna.pl

Journal of Applied Computer Science Methods

Contents