Study of Deep Neural Network and Support Vector Machine in Branch Prediction

Royan Tuscano
Department of Electrical and Computer Engineering
University of Arizona
Tucson, 85719
rtuscano@email.arizona.edu

Abstract

We have seen lot of progress in machine learning and deep neural network from year 2006.They have shown excellent performance in computer vision and Data Analytics. In this paper, we study the feasibility of the machine learning algorithm support vector machine and Deep neural networks in branch prediction. We study the branch prediction as a two-class classification problem. We compare the results of our branch prediction analysis with Deep neural network and Support vector machine using different hardware storage . We also observe that support vector machine can be used in branch prediction and can offer good results. We have also done design space exploration for the best local history register, global history register for deep learning classifier and Machine learning.In our analysis, Deep learning classifier performed better than Support Vector Machine algorithm.But the Deep learning classifier has more implementation overhead compared to the Support vector machine. Also, we learned that the hardware of 10 Kbits performance almost identical to the hardware of 1.6 Mbits, this hardware is used for storing entries of local history.We have used incremental learning in both the classifier to predict the branches.

1 Introduction

The modern processor uses up to 20 pipelines. Hence a control hazard takes place when next fetched instruction differs from the correct outcome of the branch. The control hazard affects the performance of the processor, i.e the misprediction will affect the processor performance and power consumption. The current trend is moving towards deeper pipeline, energy efficiency and smaller silicon space on the chip. Hence, the branch prediction has become an important component of the processor.

TAGE and perceptron-based branch predictors are the winners of the last five Championship Branch Prediction (CBP) competitions, they are considered to be the State-of-the-art.State-of-the-art conditional branch predictors exploit several different history lengths to capture correlation from very remote branch outcomes as well as very recent branch history. Several different approaches including perceptron-based prediction proposed to compute final prediction from neural network embedded multiple components but face the issues of increasing accuracy at the expense of extra memory usage. Hence given the current advancement in machine learning it is interesting to observe and analyze the efficiency of the machine learning and deep neural network algorithms in branch prediction.

In this paper, we seek to explore the feasibility of applying machine learning and deep neural network in branch prediction.We see branch prediction as a two-class classification problem with 0 as not taken and 1 as branch taken. We do the incremental learning on the Deep neural network and Support vector machine to observe the efficiency of the branch predictor. We use the global history, local history, and address as the feature set. We observe that the deep neural network performs better than the Support

ECE523: Engineering Applications of Machine Learning and Data Analytics (Sp2018).

vector machine in the performance. But the overall hardware overhead for deep neural classifier will be greater than the support vector machine. By comparing the performance of the deep neural network classifier, we confirm that it is a good branch predictor. We test our predictors with different size of the program counter(Instruction address), global history length and local history length. We analyze the effect of these features on the performance of the branch predictor and the hardware resources required. Our simulation result showed that the increase in the Local history register length, glocal address register length, and address range will not always lead to better prediction, but it will increase the total hardware budget.

The rest of the paper is organized as follows. Section 2 describes background work in the state-of-the-art predictor. Section 3 describes deep learning and section 4 describes support vector machine and section 5 explains our evaluation and methodology. Section 6 explains the results and analysis and section 6 summarizes the work.

2 State of the Art Branch Predictor

The TAGE branch predictor is considered to be the best branch predictor. One key advantage of TAGE predictors over other predictors is to use a geometric series of history lengths for prediction. It enables the predictor to explore the correlation between branches and very long history lengths while allocating most of the hardware resource to the short-length prediction components[1]. Another key aspect of TAGE is that it uses tag-matches when accessing each prediction component to reduce aliases[1].

Perceptron predictor is another type of State-of-the-art branch predictor. The perceptron uses simplest single layer neural network to predict branch from the address and the branch history. The perceptron is trained after every outcome or classification. Hence after every iteration, the weight of the predictor is readjusted.The Each entry in the table consists of the weights that are trained when we use corresponding instruction address. The drawback of the perceptron training is that it can only classify linear outcomes.

3 Deep neural network

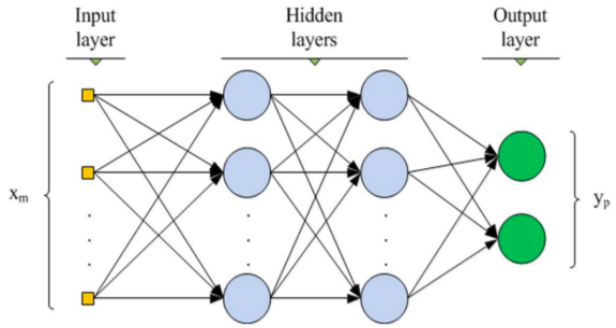


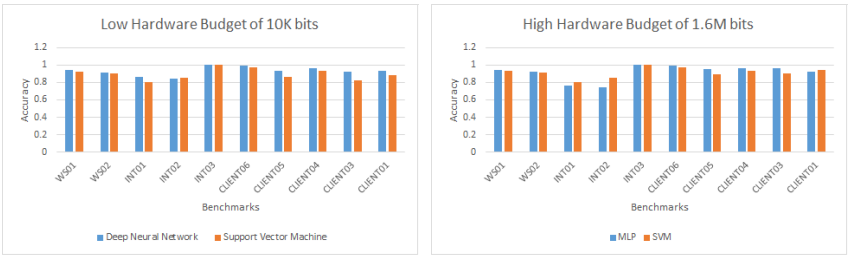
Figure 1: Deep Neural Network with 3 hidden layer

A deep neural network is an algorithm that uses a multilayer approach to extract and represent the higher-order features of the training data. The neural network uses three hidden layer and 100 nodes per layer. The input layer uses 148 and 126 input nodes for our two different deep neural network models.The output layer has 2 output nodes which represent branch taken and Not taken.We feed the input layer with the Program counter, local history register of that program counter and global history. We then compare the predicted result with correct output result and then update the global history and local history of the input program counter address.We use the Least significant bits of the address space for creating the Local history table corresponding to the addresses.

4 Support Vector Machine

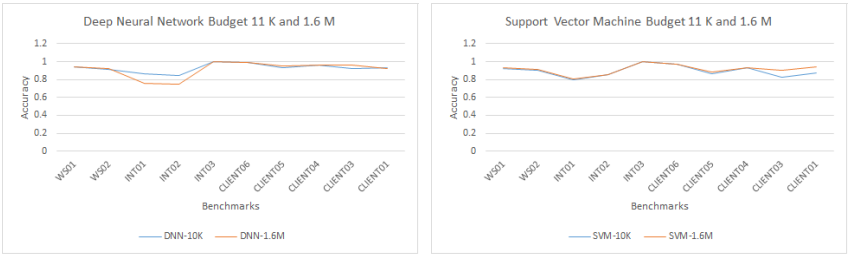
The support vector machine uses linear kernel for the two-class classification. We use l2 regularization and take program counter, Global History and local History as an input. We use incremental learning for support vector machine where we partially fit the input data. We then compare the predicted result with correct output result and then update the global history and local history of the input program counter address.We store the actual output value in the local history table corresponding to the program counter.

5 Evaluation and Methodology



(a) Hardware 10K bits (b) Hardware 1.6M bits

In order to evaluate different benchmarks. We use the branch traces obtained from Championship Branch Prediction[5].We are using the traces that run on the general purpose computer. Each benchmark consists of up to 2 million instructions. We focus on the conditional branch instruction in this traces. We use two different predictor sizes, in the first predictor uses low hardware budget of 10K bits and the second predictor uses high hardware budget of 1.6M bits.Both the predictor model uses the same Deep Neural Network and Support vector machine setup.Figure 2 shows the branch predictor with hardware budget of 10K it uses 1000 entries containing 10 bits of local history and 1 entry with 100 bits of global history which is feed into the input. We can observe that the deep neural network gives better performance in most of the benchmarks except INT01 and INT02. Figure 3 shows the performance of the Deep neural network and support vector machine on the high hardware budget of 1.6M it uses 10000 entries containing 16 bits of local history and 1 entry with 100 bits of global history. Hence we can see that the deep neural network is better suited for this low budget hardware.Here also the deep neural network performs better except the INT01, INT02, and CLIENT01. Hence we can choose deep neural network over Support vector machine. But then the deep neural network contains 3 hidden layers and around 500 nodes. Which makes it difficult to implement in the hardware. Compare that to the Support vector machine which will be relatively easy to build and will consume less power.Hence if we consider the implementation perspective of the algorithm on hardware than the Support vector machine will be relatively easy to build and consume less overhead, while providing a comparable prediction accuracy to the deep neural network.



(a) DNN (b) SVM

Figure 4 and Figure 5 represents the prediction efficiency with respect to the amount of the hardware used. For low budget DNN and SVM we use 128 input nodes ,while for high budget DNN and SVM we use 146 input nodes. We can see in both the figures that the accuracy of the DNN with low budget and high budget is almost same, in fact for benchmark INT01 and INT02 we obtain better efficiency.

Hence it is advisable to use low budget DNN and SVM over high budget , Since we get almost same prediction accuracy. But we get hardware savings of 1.5 M bits which will save cost and energy.

6 Conclusion

In this paper, we use binary classification perspective to analyze and explore branch prediction. We have applied Support Vector Machine and Deep Neural Network as classifiers. The first observation was that Deep neural network performs better than Support Vector Machine in most of the benchmarks. The second observation was that impact LHR, GHR and hashed length of Program counter on the accuracy of the branch prediction. We concluded that not always the increase in these parameters will lead to higher accuracy, but in fact, lower values of these parameters can give good results and save hardware size.At last, we also saw that Support vector machine can offer comparable performance to the deep neural network at lesser hardware implementation cost.

References

[1] Yonghau Mao, Junjie Shen & Xiaolin Gui, *A Study on Deep Belief Net for Branch Prediction*, IEEE Access, 2017

[2] Kevin Irick, Michael DeBole, Vijaykrishnan Narayanan, Aman Gayasen.*A hardware efficient implentation of Support Vector Architecture for FPGA*, Field-Programmable Custom Computing Machines, 2008

[3] *https://en.wikipedia.org/wiki/Deeplearning/*

[4] *https://www.hackerearth.com/blog/wp-content/uploads/2017/01/multi.jpg*

[5] *https://www.jilp.org/jwac-2/*

[6] Jichen Wang, Jun Lin, Zhongfeng Wang.*Effiecient hardware architectures for deep convolution neural network*, IEEE Transactions on Circuits and Systems, 2017