Branch prediction with Machine Learning

Strahinja Praska
Faculty of Technical Sciences, Novi Sad, Serbia

Abstract

Branch prediction is a critical aspect of modern pipelined processors, aimed at improving instruction throughput by guessing the direction of branches in code. This paper explores the use of perceptron-based branch predictors to enhance prediction accuracy. The implementation involves a simulation of an instruction pipeline in Rust, integrating a custom branch predictor. Results indicate that perceptron-based predictors can significantly reduce misprediction rates across various datasets, with optimal history lengths ranging between 12 and 62.

Introduction

Instruction pipelining is a technique used in RISC processors to process multiple instructions simultaneously by dividing the task into several stages: Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Writeback (WB). Without pipelining, it would take 5n cycles to run n instructions, as each instruction would wait for the previous one to complete. Pipelining improves throughput, enabling the processor to handle one instruction per cycle. However, branch mispredictions can introduce significant delays. Branch predictors aim to guess the direction of branches to maintain the flow of instructions. This paper focuses on perceptron-based branch predictors due to their ease of hardware implementation and interpretable decision-making process.

Methodology

Perceptron-Based Branch Predictor

Motivation

Perceptrons are chosen for their straightforward hardware implementation and understandable decisions. A single-layer perceptron is used, consisting of one artificial neuron connecting several input units by weighted edges to one output unit. The perceptron learns a target boolean function t ofn inputs, represented by bits of the global branch history register. The function t predicts whether a branch will be taken or not.

Training

The perceptron is represented by a vector of weights (signed integers). The output is the dot product of the weight and input vectors. The perceptron is retrained after each known outcome, with the training algorithm adjusting the weights based on the branch outcome. The training continues until the output meets a specified threshold θ.

Linear Separability

Perceptrons can only learn linearly separable functions. The solution to the equation

w0+∑xiwi=0
w0 +∑xi wi =0 is a hyperplane that separates the inputs for which the perceptron predicts true from those for which it predicts false. This limitation is acceptable for the branch prediction task.

Construction

The branch predictor is constructed using a table of N perceptrons stored in fast SRAM. The high-level operation involves retrieving the corresponding perceptron by address, computing the dot product with the global history register, predicting the branch direction, and updating the perceptron weights based on the actual outcome.

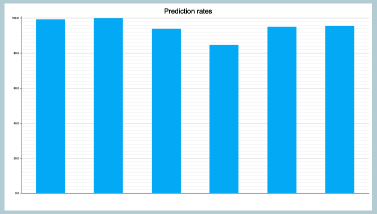
Implementation and Results

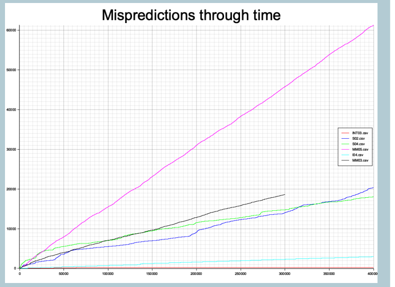
The project is implemented as a Rust simulation of an instruction pipeline with a custom branch predictor. The results indicate that the optimal history length for branch prediction ranges from 12 to 62. A simple feed-forward neural network is also tested for comparison. The prediction rates for various datasets show significant accuracy, with some datasets achieving over 99% prediction rates.

Conclusion

Perceptron-based branch predictors offer a robust solution for improving branch prediction accuracy in pipelined processors.
The implementation in Rust demonstrates their practical applicability and efficiency.
While perceptrons are limited to linearly separable functions, their performance in branch prediction tasks is satisfactory. Further exploration into neural network-based predictors could provide additional insights and potential improvements in prediction accuracy.

Results





References

Dynamic Branch Prediction with Perceptrons, Daniel A. Jimenez, Calvin Lin

https://www.kaggle.com/datasets/dmitryshkadarevich/branch-prediction

https://en.wikipedia.org/wiki/Branch_predictor

https://web.archive.org/web/20131227033204/http://hpc.serc.iisc.ernet.in/~govind/hpc/L10-Pipeline.txt}