

Perceptron Inspired Branch Prediction



“ Prediction is very difficult, especially about the future. ”

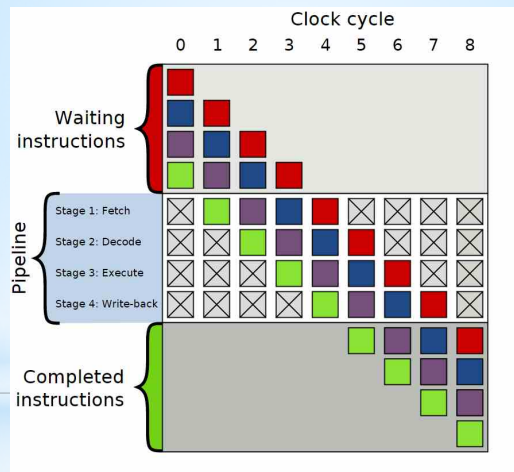
Robert Storm Petersen (1882-1949)
Danish cartoonist, writer, animator,
illustrator, painter and humorist

By David Egolf, for CPTR 350. March 10, 2016.

Presentation Overview

- Motivation
- Transition: saturating counters
- Single perceptron predictor
 - Motivation
 - Derivation
 - Example
- Conceptual overview of refinements

Motivation

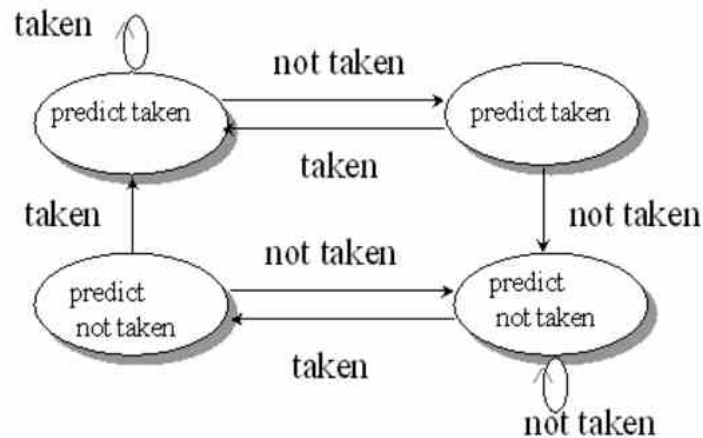


Conditional branching instructions cause problems in pipelines.

Motivation

- Mispredictions lead to...
 - Wasting work on wrong instructions
 - Stalling while grabbing new instructions
- We want a predictor that is...
 - Accurate
 - Fast (simple)
 - Low power

An attempt: Saturating Counter



An attempt: Saturating Counter

- Use context specific counters
- Keep track of n history bits
 - Add one counter per possible histories
 - Problem: 2^n counters required

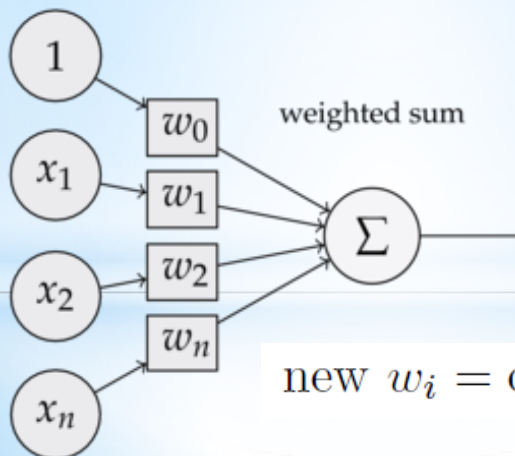
Times Ago	Branch or Not? (Y/N)
1	Y
2	N
3	N
4	N
5	Y

A Big Problem

- We want to use history in predictions
- But, avoid exponential growth of hardware
- What to do?

Single Perceptron Predictor

inputs weights



Example

- Predict user provided pattern
 - [1, -1, -1, 1, 1]
- Using 15 history inputs
 - There are 16 weights
- Error disappears quickly!

```
>> perceptronDemo
Predicted Actual Error
-1 1 -2
1 1 0
-1 -1 0
1 1 0
1 -1 2
-1 -1 0
1 1 0
1 1 0
-1 -1 0
1 1 0
-1 -1 0
-1 -1 0
1 1 0
1 1 0
-1 -1 0
1 1 0
-1 -1 0
-1 -1 0
1 1 0
1 1 0
-1 -1 0
1 1 0
-1 -1 0
-1 -1 0
```

Problems with Perceptrons

- Calculating the weighted sum is slow
- Linear separability issues

Refinement: Path Based

- Decrease latency
- Increase precision

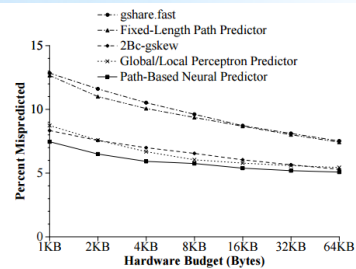


Figure 6. Average misprediction rates per hardware budget

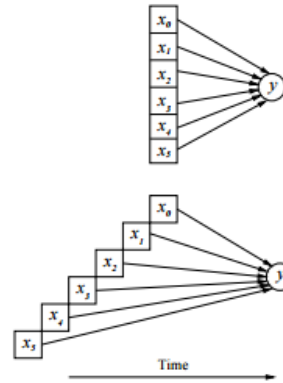
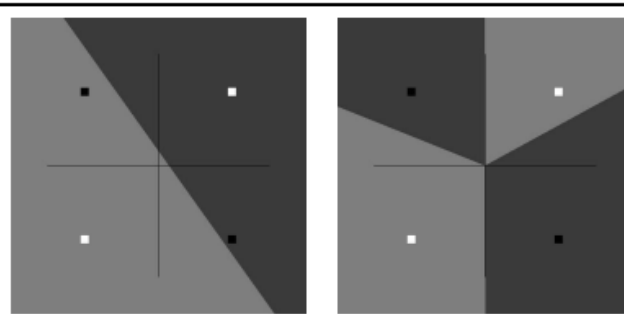


Figure 1. Rather than being done all at once (above), computation is staggered (below).

Refinement: Piecewise Linear

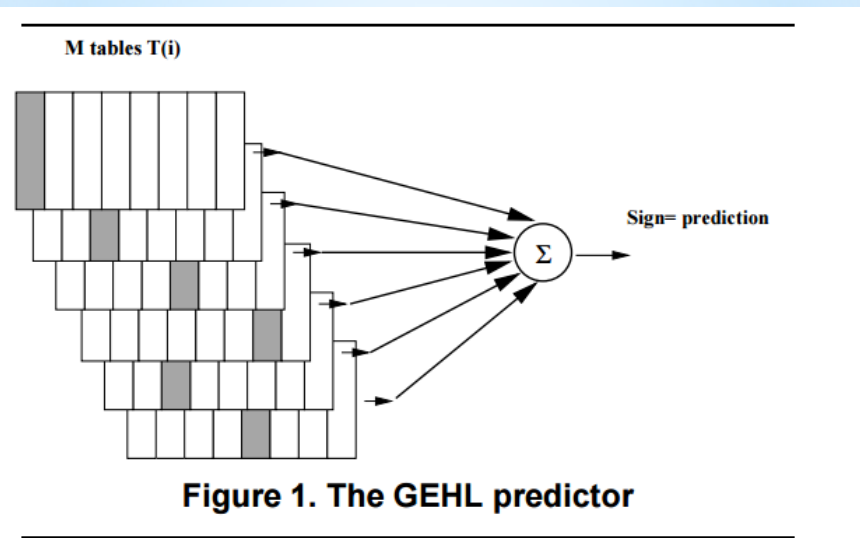


(a)

(b)

Figure 1. The XOR function cannot be learned by a perceptron (a), but can be learned using a piecewise linear decision surface (b).

Refinement: O-GEHL



Review of Presentation

- Why accurate branch prediction?
 - Avoid stalling
 - Avoid doing wasted work
- Why perceptrons?
 - Simple, scale well, accurate

Review of Presentation

- Problems with perceptrons?
 - Slow, linear separability issues
- Some ways to improve:
 - Path based prediction
 - Piecewise linear prediction
 - O-GEHL

Questions?