# **Discussion of Branch Prediction**

이 문서는 Project 3 과제를 끝마치고 유 교수님께서 제시하신 추가적인 Branch Prediction에 대한 나의 생각을 정리해 놓은 문서이다. Fibonacci 프로그램의 경우, Branch Prediction을 하기가 많이 어려운데, 그 이유는 Branch Prediction에 대한 Entropy 정도가 높은 프로그램이어서 그렇다. 실제로 수업시간에 배운 Branch Prediction 기법들을 사용한다면, Fibonacci 프로그램은 다른 프로그램들에 비해 더 낮은 Branch Prediction 성공률 (50 ~ 70%)을 보인다. 유 교수님께서 제시하신 의문은 이러한 Branch Prediction에 대한 Entropy가 높은 프로그램에 좋은 성능을 보이는 Branch Predictor를 만들 수 있는 가였다.

가장 먼저 떠오른 방법은 Perceptron 이론을 적용하는 방법 이였다. 이는 Global Share 기법을 착안하여 생각해낸 방법이다.

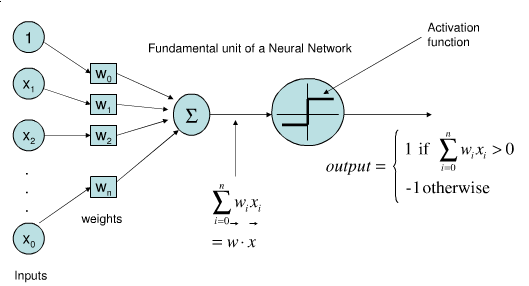


Figure 1 - Perceptron (Jupyter &amp; Perceptron)

Figure 1에서 확인할 수 있듯이, Perceptron에서는 X0 ~ Xn까지의 값들에 가중치를 곱하고, Bias를 설정하여 이를 모두 더한 값이 임계치를 초과라면 1을, 아니면 0을 출력하는 형태이다. 이 방법을 적용한 Branch Predictor는 다음과 같이 작동될 것이고 생각했다.

1. Global History Register (GHR)의 비트 수가 8 비트라고 가정해보자. 그렇다면 GHR은 이전 8개에 대한 Branch Prediction History를 저장하고 있을 것이다.
2. 여기서 최근 Branch Prediction이 더 높은 Prediction 연관성을 보인다고 가정한다면, 가중치 W는 W0=8/36, W1=7/36, …, W7=1/36로 설정할 수 있다.
3. 여기에 Branch Prediction이 항상 발생한다고 가정하고 Bias를 1로 설정한다.
4. 그리고 기존의 Global Share Predictor의 방식으로 Branch Prediction을 수행해 주면 된다.

이와 같은 방식으로 Branch Predictor를 만들면 제시된 프로그램의 Branch Prediction 정확도를 높일 수 있겠 다라고 생각하였으나, 이 방법 역시 결국에는 Entropy와 무관하게 예전의 Branch Prediction 결과로 예측하는 거라 합리적인 방법이 아니라는 생각이 들었다.

지금 학기 중에 할 수 있는 생각은 여기까지 인 것 같다. 차후에 시간이 남거나 방학 중에 체코에 있을 동안 한번 생각해보기 좋은 주제인 것 같다.

# **Citation**

* Jupyter &amp; Perceptron. (n.d.). Retrieved May 22, 2022, from <http://incredible.ai/jupyter/2016/05/02/Jupyter-Perceptron/>
* 1.10. decision trees. scikit. (n.d.). Retrieved May 22, 2022, from <https://scikit-learn.org/stable/modules/tree.html>