Use Case White Paper

Financial Industry Use Case:
Public Sentiment Extraction (NLP, Word2Vec)

OVERVIEW

This paper is an overview of using Natural Language Processing (NLP) in the context of predicting stock movement. It indicates the cost savings using Intensivate technology to be **76% versus state of art servers, and 67% compared to state of art GPUs**.

NLP

Natural Language Processing (NLP) is often used in the Financial Sector to discover public sentiment on various topics by continually processing news feeds, twitter, etc.

EXTRACT SENTIMENT

sentiment about…

- a company
- a market sector
- overall market
- key indicators

SIGNIFICANT EVENTS

such as…

- oil spill in the Gulf
- key legislation passed
- accounting scandals
- employee layoffs

This data translates into consequences on stock prices

This kind of analysis is powered by specialized NLP algorithms. One of the most popular is called Word2Vec, which was invented by Google. Word2Vec has been highly optimized on a number of hardware platforms, including Intel Xeon CPUs (97% of all servers have Xeon CPUs inside), and also NVidia GPUs.

Word2Vec

To use Word2Vec, you first train it. You have a fixed set of examples of the right thing to do, and then run that set through your system again and again. Each time the system gets a bit better, until it flattens out. In order to get a given accuracy, you have to repeat the process a certain number of times. The number of times you repeat sets the total amount of work done to train the system.

Sentiment analysis on Twitter using word2vec and keras
https://ahmedbesbes.com/sentiment-analysis-on-twitter-using-word2vec-and-keras.html
Sentiment analysis using word2vec
https://www.kaggle.com/varun08/sentiment-analysis-using-word2vec

Training Word2Vec

Training takes a long time! There is a competition on who can do training the best. Two measures are accuracy and time to complete. But in the era of The Cloud, users are charged by the kind of hardware multipled by how long that hardware is used. More expensive hardware runs faster, but it also costs more per hour! In the end, the total amount of work is determined by the accuracy. So, the ultimate measure is how much do you spend for the desired accuracy?

A study was done on AWS, to answer just this question. It tried many different variations, looking for the lowest cost way to get the desired accuracy. It turned out to be a tie, between Xeon CPUs that use a special extension called vector units (AVX2) and the powerful P100 GPUs from Nvidia. Both cost almost the exact same to complete the training and get nearly the same accuracy.*

Word2Vec with Intensivate

At Intensivate, we have run the same Word2Vec software, on the same training set, and measured the ratio of performance. IntenCore processed the training set at the rate of 70,000 training samples per second, compared to 110,000 on AWS state of art Xeon CPUs. When total cost of ownership is factored in, Intensivate delivers the same accuracy, running the same source code on the same training set, at **76% less cost to arrive at the final, trained system**.

The same experiment was done with P100 GPUs from Nvidia. Versus those, **the savings was 67%**.

This means that processing twitter, Reuters, and all the news feeds in existence for the impact on stock prices can be done at a savings of 67% compared to state of the art GPUs and a savings of 76% compared to state of art servers.

Floating Point Math Intensive

An important note here is that Word2Vec is floating point math intensive, and Xeon CPUs have a special co-proccesor that is just for such floating point intensive applications. It's called a "vector extension" denoted AVX. Using this extension requires software to be hand-tuned and use special software constructs, as mentioned here. In contrast, Intencore does not need any special software tweaks in order to deliver high performance. On Word2Vec, it does not use the special Intel directives, but only uses normal C++ code, and still delivers 85% of its peak performance.

* **Amazon SageMaker BlazingText: Parallelizing Word2Vec on Multiple CPUs or GPUs**
https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-blazingtext-parallelizing-word2vec-on-multiple-cpus-or-gpus/