Connectionist and Other Computation Models

Jiawei Hong and Xiaonan Tan
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 88-100

September 1988

Connectionist and Other Computation Models

Jiawei Hong and Xiaonan Tan
Department of Computer and Information Science
university of Massachusetts at Amherst

September 1988

Abstract

In this paper we present the analytical results about the computational potential and limitations of connectionist models. The well known connectionist models defined by an $n \times n$ real weight matrix can be simulated by a non-uniform aggregate of $O(n^3 log\, n)$ Boolean gates with a slow down factor of $O(log\, n)$. Therefore the connectionist model is similar to all other non-uniform computation models. We develop the current connectionist model into the asynchronous model and the superconnectionist model. The asynchronous connectionist model, the superconnectionist model as well can be simulated by a parallel machine with polynomial space and polylog time slow down. So both of them are also similar to all other non-uniform computation models. Our work builds a bridge which connects connectionist models to the conventional ones and explores connectionist models in more depth.

1 Introduction

In 1985, when the Artificial Intelligence journal was publishing its 25th volume, the editors constructed a set of 10 questions and sent to many experts in AI area [1]. The question 1 was: A.I. has developed rapidly in the last decade. What do you feel have been the most significant scientific and/or technological advances in that time scale? Of the dozen or so specific topics which people thought were significant, only two were mentioned more than twice: connectionism (massive parallelism) and expert system.

Some people mentioned that connectionist systems would be an exciting recent idea with great promise. Because they believe that many of the core problems of AI, such as pattern matching, context sensitivity, representation of real world knowledge and plausible inference, are better approached on a non-symbolic computational basis. Massively parallel connection networks, spreading activation and evidential inference appear to be a natural scientific paradigm for exploring fundamental questions of intelligence and to have close correspondence to evolving understanding in psychology and neuroscience.

As people have expected, the rate of progress along connectionism line is great. Connectionist techniques are now widely applied to the complex processing needs of "high-level cognitive processing". However, very little is known, as yet, about the computational potential and limitations of connectionist machines. The relationship between connectionist model and existing computation models is not well understood. More theoretical work needs to be done. Motivated by this, we make an analytic comparison of connectionist model to the conventional computation models. We attempt to reveal the deep relation among them and to build a bridge that connects connectionist model to the conventional ones.

The first author has proved that all the conventional computational models are not only equivalent in Turing's sense, but also similar in the sense that for the same problem class they need essentially the same parallel time, essentially the same space and essentially the same sequential time simultaneously [2, 3]. By essentially the same we mean polynomially related. This is called similarity principle. Therefore the problem 1 in this paper is: Does the similarity principle hold good for connectionist model? The answer is yes. We will prove that the connectionist model is similar to all other non-uniform computation models. The connectionist model is certainly well suited for handling specialized problems in perception; it leads us to think about new ways of representing knowledge, new ways of organizing problem solving and so on. However, like all computer systems, the connectionist model still can not help to solve the problem of combinatorial explosion usually encountered in A.I. problems.

As a novel computation model, the connectionist model has great potentialities to develop. We develop the current synchronous connectionist model into the asynchronous connectionist model which is much stronger and again into the superconnectionist model which is extremely strong. We prove that the asynchronous connectionist model, more surprisingly, the superconnectionist model as well, are similar to all other non-uniform computation models. They can not overcome the difficulties of NP-completeness or NP-hardness either.

The theoretical results presented in this paper build a bridge between the connectionist models and other computation models and point out the limitation. This is not to say that we should be pessimistic. In fact, the connectionist models are appropriate for AI problems. We do not believe that a human brain can solve combinatorial explosion, therefore we do not think AI should worry about NP-hardness too much. We do not see evidence for, nor can we imagine, limits to AI research possibilities.

For convenience' sake, we use the following terminologies. We say a problem class can be solved in a reasonable time (reasonable parallel time or reasonable space) if it can be solved in polynomial time (polynomial parallel time or polynomial space). While polynomial complexity means reasonable, polylog complexity means very efficient. We say a problem class can be very efficiently solved in parallel, if this problem class can be solved in polynomial space and polylog parallel time simultaneously (the problem is in NC, as stated in theory). We say two models are similar if these two models can simulate each other in such a way that both parallel time and space are polynomially related.

2 Connectionist Models

A connectionist machine consists of a set of n formal neurons, $Neuron_i, i = 1, 2, ...n$, connecting to each other through nerves. Mathematically, we assume that the nerve connecting the ith neuron to the jth neuron is $Nerve_{ij}$, which is weighted by a real number w_{ij}. Each neuron has a threshold, a real number. The threshold of the jth neuron is denoted by T_j. At each time, a neuron may be in either an **on** state or otherwise an **off** state. we use a binary function $v(i,t)$ to represent the state of the ith neuron at time t, i.e., if the ith neuron is on at time t then $v(i,t) = 1$, otherwise $v(i,t) = 0$. If the ith neuron is on, then it will affect the jth neuron through $Nerve_{ij}$ weighted by a real number w_{ij}. The jth neuron sums all the weights where the neuron is on. If the total sum is larger than its threshold T_j, then at time $t+1$ the jth neuron will be on, otherwise the jth neuron will be off, as stated by the following formula:

$$v(j,t+1) = \begin{cases} 1, & if \sum_{i=1}^{n} w_{ij}v(i,t) > T_j, \\ 0, & otherwise. \end{cases} \quad (1)$$

We may put in the system another neuron, the $n+1th$, which is always on and disconnected from any one else. Obviously the system is equivalent to the original one. Now we add a nerve connecting the $n+1th$ neuron to the jth neuron. We put a weight $-T_j$ on it and change the threshold of the jth neuron from T_j to 0. Because the $n+1th$ neuron is always on, the behavior of the jth neuron will remain the same as before. Therefore later we need only consider the case that all the threshold numbers are 0: $T_j = 0$ for all j. Thus, a connectionist machine of size n can be represented by a real matrix w_{ij} of size $n \times n$.

Compared with other conventional computation models, the above model is very novel. A Turing machine can be represented by its transition function, which is a finite table. A RAM or a PRAM machine can be determined by its program, which is a finite string. The total information content is bounded by a constant (does not depend on the input length n). We call this kind of computation model uniform model. If the total information content is increasing and bounded by a function of n, then we say this computation model is non-uniform. But the connectionist model is different. A connectionist machine can be given by a matrix having $n \times n$ real numbers which could be infinitely long.

It is not difficult to see that the connectionist model can simulate other computation models, such as an aggregate. Roughly speaking, an aggregate is a collection of many Boolean gates connecting to each other. All these gates work synchronously in an obvious way. At the very beginning of the automata theory, people knew that any Boolean gate could be simulated by formal neurons. Because all conventional computation models are similar, if connectionist model can simulate aggregate, then it can simulate other conventional computation models with essentially the same efficiency.

But on the other hand, can the aggregate model simulate the connectionist model? In this section we will prove the following theorem.

Theorem 1 *A connectionist machine of n neurons defined by a real matrix of $n \times n$ real numbers can be simulated by an aggregate consisting of $O(n^3 \log n)$ Boolean gates with time slow down only $O(\log n)$. Connectionist model is similar to all other non-uniform computation models.*

The key point to prove the above theorem is a gap theorem which guarantees that though an arbitrary real number could be infinitely long, we need only $O(n \log n)$ bits for each weight to define an arbitrary connectionist machine of n neurons. The rigorous proof for n dimensional case of the gap theorem can be found in [4]. A simple proof of 2D case can be found in the appendix of this paper, which gives the reader a good intuition.

We cite the gap theorem here.

Theorem 2 *(Gap Theorem 1) Assume that in n dimensional space we have many hyperplanes:*

$$P_\alpha : \sum_{i=1}^{n} a_{\alpha i} x_i = b_\alpha, \alpha \in S,$$

where S is an arbitrary set and $a_{\alpha i}$, b_α are integers satisfying $\sum_{i=1}^{n} |a_{\alpha i}| \le K$ for some constant K. Then these hyperplanes divide the n dimensional space into many regions. Some regions are degenerate, meaning that the volume of the region is 0. Otherwise the region is non-degenerate. If a region is non-degenerate, then we can inscribe an n dimensional ball inside, such that the radius d of the ball is at least

$$d \ge \frac{1}{K^{n+1}(n+1)}.$$

Theorem 3 *(Gap Theorem 2) If all $b_\alpha = 0$, then the hyperplanes divide the n dimensional space into many cones. If a cone is non-degenerate, then we can find an integer point inside the cone such that the binary length of the coordinates of the integer point is bounded by $\lceil log(K^{n+1}(n+1)+1) \rceil$.*

Now we prove theorem 1. To simulate the connectionist model by the aggregate model, we should find some integers m_i to simulate the real weights w_i such that

$$\sum_{i=1}^{n} w_i v(i,t) \ge 0 \tag{2}$$

if and only if

$$\sum_{i=1}^{n} m_i v(i,t) \ge 0 \tag{3}$$

for all possible $v(i,t)$. Anyhow, the value of $v(i,t)$ is either 0 or 1. The point $(w_1, w_2, ...w_n)$ is inside a cone divided by the hyperplanes:

$$\sum_{i=1}^{n} v(i,t)x_i = 0, \tag{4}$$

for all possible values of $v(i,t)$.

We can find an integer point $(m_1, m_2, ..., m_n)$ in the same cone. Since they are in the same cone divided by (4), inequality (2) holds if and only if inequality (3) holds. Since $v(i,t)$ is either 0 or 1, the summation of the coefficients (the $\sum_{i=1}^{n} |a_{\alpha i}|$ in Gap theorem 1) is bounded by n (the constant K in Gap theorem 1 is n):

$$\sum_{i=1}^{n} |v(i,t)| \leq n.$$

The binary lengths of coordinates of the integer point $(m_1, m_2, ..., m_n)$ are bounded by

$$\lceil log(K^{n+1}(n+1)+1) \rceil \leq \lceil log(n^{n+1}(n+1)+1) \rceil \leq \lceil log(n+1)^{n+2} \rceil \leq \lceil (n+2) log(n+1) \rceil.$$

Therefore $\lceil (n+2)log(n+1) \rceil$ bits will be enough to define a connectionist machine of n neurons. Since we have the conclusion that $\lceil (n+2)log(n+1) \rceil$ bits will be enough for each weight, we can use a standard parallel adder to add all the weights together. There are n numbers, each of length $\lceil (n+2)log(n+1) \rceil$. The total parallel time needed is $O(log\, n)$ (the slow down factor). The total number of gates to simulate one neuron is $O(n^2 log\, n)$. Therefore the total number of Boolean gates in the aggregate is $O(n^3 log\, n)$.

The space used by both machines are obviously polynomially related. Since we can assume that the (parallel) time is at least proportional to the logarithm of the space, the parallel time used by both machines are also polynomially related.

3 Asynchronous Connectionist Model

The connectionist model described in the previous section is actually the synchronous one, where all neurons work synchronously. It is instructive to extend the connectionist model to an asynchronous one because working asynchronously seems to be the way that the brain gets jobs done. There are two kinds of meanings for asynchronism. The weak sense of asynchronism means that the computation result is always the same while the time schedules of the elements vary at random. Obviously, this kind of asynchronism does not help the computation. The strong sense of asynchronism means that the system is deterministic and the time schedules (for example, the time delays) of the computation elements are some real numbers rather than integers. (For example, a signal can be sent from a gate to another exactly in π second.) We consider the case of the strong asynchronism. Merely changing the time schedules of the elements brings the synchronous connectionist model to the asynchronous one. We define an asynchronous connectionist model as follows:

We assume that there are n neurons as before. From the ith neuron to the jth neuron there is a nerve, $Nerve_{ij}$. This nerve has a time delay d_{ij}, an arbitrary real number. The information that the ith neuron is off or on (the value $v(i,t)$ is 0 or 1) will be weighted by a factor w_{ij}, delayed by a time equal to d_{ij}, and arrive at the jth neuron. The jth neuron sums the total delayed and weighted values among all its neighbors during the (continuous) time. If the summation is steadily greater then 0 (or steadily not greater than 0) during a period longer than δ_j, then the jth neuron should be on (or off). The number δ_j is an arbitrary real number and is called the time threshold of the jth neuron. All these time delays and time thresholds satisfy the following condition:

$$0 < \delta \leq \delta_j \leq \Delta \tag{5}$$

$$0 < \delta \leq d_{ij} \leq \Delta \tag{6}$$

where δ and Δ are two constants.

We attempt to probe deeply into the asynchronous connectionist model.

The history of a neuron during a period is a list of the following form:

$$(t_1,\ 0;\ t_2,\ 1;\ \ t_3,\ 0;\ \ t_4,\ 1; \ldots)$$

meaning that from time t_1 on the value of the neuron is 0, from time t_2 on, the value of the neuron is 1,..., and so forth.

It is not difficult to show that if we know the histories for all neurons during time period $[t - 2\Delta, t]$, then we can compute histories for all neurons during time period $[t - 2\Delta + \delta, t + \delta]$. The length of the time period is 2Δ, therefore there are only constant many terms in the history list. Of course, these numbers $t_1, t_2...$ might be arbitrary real numbers. If they are integers (later we will prove that all these numbers can be simulated by integers of length bounded by a polynomial of n), then we can think the history of a neuron during time period $[t - 2\Delta, t]$ as the state at time t. If we know the states for all neurons at time t, then we can compute the states of all neurons at time $t + \delta$. (In this way, we can treat the asynchronous connectionist model as a synchronous one.) This computation can be very efficiently solved in parallel. In other words, this problem can be solved in polynomial space and polylog parallel time (in other words, the problem is in NC).

If we change the time delays a little bit, the result of the computation might be changed, because a signal A now may arrive before signal B but it originally should arrive after signal B. But if we change the time delays and time thresholds in such a way that all time orderings remain unchanged, then the result of the computation will be the same. Any time ordering can be expressed by a form as

$$\sum_{ij} a_{ij} d_{ij} + \sum_{j} b_j \delta_j > 0$$

where a_{ij}, b_j are some integers satisfying

$$\sum_{ij} |a_{ij}| + \sum_{j} |b_j| \leq t,$$

where t is the total computation time.

By our gap theorem again, we can simulate the real numbers d_{ij}, δ_j by integers. Since the total number of variables (the dimension) is $n^2 + n$ in stead of n, and the summation of the absolute values of the coefficients is bounded by t in stead of K, we know that the length of these integers found is bounded by

$$O(n^2 log\, t).$$

The computation time t can not be much higher than an exponential function of n, otherwise the computation will fall into an endless loop. Therefore $log\, t$, and the length of these integers found, is bounded by a polynomial of n.

We summarize the significant properties of the asynchronous connectionist model as follows:

1. The state of a neuron at time t (the history of the neuron during time period $[\ t-2\delta, t\ ])$ can be represented by a binary string of length n^* ($*$ is a constant, therefore n^* is a polynomial of n).

2. The states of all neurons at time $t+\delta$ can be computed from the states of all neurons at time t. The computation is in NC (can be very efficiently solved in parallel).

Based on these two points only, we can prove the following theorem (the proof will be given in the next section).

Theorem 4 *The asynchronous connectionist model is similar to all other non-uniform computation models.*

Therefore neither the asynchronous can help to solve the combinatorial explosion encountered with A.I. problems.

4 Superconnectionist model

The fundamental importance of connectionist model lies in reflecting architectural considerations of the brain, building bridges to neurobiology. We may try to understand the architecture of the mind better by continuously developing the connectionist model.

We have extended a connectionist model from a synchronous model to an asynchronous one. In this section, the connectionist model will be strengthened in a tremendous way. We call the developed model superconnectionist model.

A superconnectionist model is a set of n neurons connecting to each other. At any time t, a neuron is in a state in S. At time $t+1$, the state of neuron i is determined by the states of all neurons and a (may be different) mapping

$$f_i : \ S^n \longrightarrow S.$$

And we have the following assumptions for the superconnectionist model:

Assumption 1 *The size of set S is bounded by 2^{n^*}.*

Here $*$ means a constant again, like 2 or 10, for example. This assumption says that the state of a single neuron can be represented by a binary string of length bounded by a polynomial of n. As mentioned in the last section, the asynchronous connectionist model satisfies this assumption. This assumption is very reasonable. The bound is large enough. Remember that n is the total number of neurons in the superconnectionist model. For example, we have 10^{12} or even more neurons. Let us say $*$ is 2. The fact that a neuron have more than 2^{n^2} states means that a single neuron can remember more than 10^{24} binary bits!

We assume that the functions of different neurons are different. We did not put any restriction on the mapping f_i. But if the mapping is too strong, such as NP-hard or even non-recursive, then the model is not interesting. Therefore we give another assumption:

Assumption 2 *The mapping f_i is in NC. That is, the mapping can be realized in polynomial space and polylog parallel time.*

This assumption is also reasonable, because we can not expect that a single neuron can instantly solve a computation problem that can not be solved very efficiently by a parallel computing system. As mentioned before, the asynchronous connectionist model satisfies this assumption. Therefore the asynchronous connectionist model is a special case of the superconnectionist model.

It is apparent that the superconnectionist model is very strong. The limits put by Assumptions 1, 2 are in fact negligible. So far it is hard to image that the real neurons in the brain would be functionally beyond the above assumptions.

Surprisingly enough, even though the superconnectionist model looks enormously powerful, we can still prove that it is similar to all other non-uniform computation models.

Theorem 5 *The superconnectionist model is similar to all other non-uniform computation models. A superconnectionist model of n neurons can be simulated by a parallel machine within reasonable space (polynomial space) very efficiently (with polylog time slow down). Their information complexities are the same.*

This theorem is not difficult to prove. Since each mapping $f_i, i = 1, 2, ...n$ is in NC, by definition there is a circuit C_i of polynomial width and polylog depth to accomplish the job. The whole design of the parallel computer to simulate the superconnectionist model is shown in Figure 1. Obviously, the width of the machine is n times a polynomial of n, still a polynomial of n. This machine can simulate one step of the superconnectionist model by accomplishing one cycle. The time slow down in the simulation is a polylog function of n, and therefore the parallel time of the simulating machine is also polynomially related with the parallel time of the superconnectionist model.

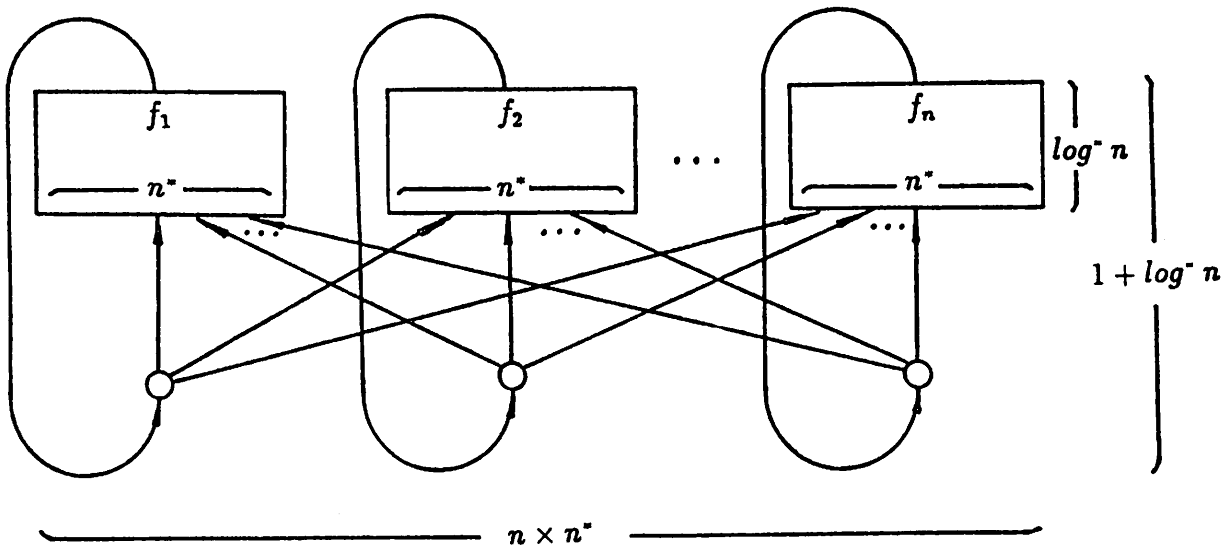

Figure 1.

Since the asynchronous connectionist model is a special case of the superconnectionist model, Theorem 2 is also valid.

5 Conclusion

The connectionist model consisting of neuron networks spreading activation and evidential inference is of fundamental importance. It reflects architectural considerations of the brain and exploits massive parallelism. It leads us to think about new ways of representing knowledge, new ways of organizing problem solving and in particular new ways of thinking about relatively low level perceptual and memory mechanisms, and so on.

To understand the connectionist model in a deeper way, we analyse the computational potential and limitations by comparing it with other computation models; we develop the current connectionist model into the asynchronous model and the superconnectionist model. Our theoretical results indicate that the connectionist model of n neurons can be simulated by a non-uniform circuit of $O(n^3 log\, n)$ Boolean gates with slow down factor of $O(log\, n)$, the asynchronous connectionist model and superconnectionist model can be simulated by a parallel machine within reasonable space (polynomial space) very efficiently (with polylog time slow down). This means that the connectionist models are similar to all other non-uniform computation models. Our work builds a bridge that connects connectionist models to the conventional computation models and points out that like all computer systems, the connectionist model still can not help to solve the problem of combinatorial explosion usually encountered in A.I.

However, we can hardly imagine that a brain cell would be functionally stronger than a formal neuron defined in the superconnectionist model. If we agree that the function of the brain is the outcome of the functions of all its neuron cells, then the brain can not solve an NP-hard problem either. Therefore, we do not think that the problem of combinatorial explosion should be the real limit to AI.

References

[1] Bobrow, D. G. and Hayes, P. J., "Artificial Intelligence – Where are we?" *Artificial Intelligence*, 25(1985)375-415.

[2] Hong, Jiawei, "On Similarity and Duality of Computation," *FOCS 1980 and also Information and Control*, 1985.

[3] Hong, Jiawei, *Computation: Computability, Similarity and Duality*, Pitman Publishing Ltd., London, 1986.

[4] Hong, Jiawei, "On connectionist models," *Communications on Pure and Applied Mathematics*, to appear 1988.

6 Appendix

The Proof of the Gap Theorem in 2D Case

Theorem 6 *Assume that there are many lines on the plane.*

$$a_{\alpha 1}x_1 + a_{\alpha 2}x_2 = b_\alpha \qquad \alpha \in S$$

where $a_{\alpha i}$ and b_α are some integers satisfying

$$|a_{\alpha 1}| + |a_{\alpha 2}| \leq K$$

for some constant K. They divide the plane into many regions. If a region is non-degenerate, we can inscribe a disc inside, such that the radius d of the disc is at least $1/3K^3$.

Proof. Assume that the radius of the disc is d and its center is $\overline{y} = (y_1, y_2)$, that a line

$$a_1x_1 + a_2x_2 - b = 0$$

is tangent to this disc and the tangent point is $\overline{z} = (z_1, z_2)$. We use $N = \sqrt{a_1^2 + a_2^2}$ to denote the norm of the line. Then the unit normal vector of the line is $A_0 = (a_1, a_2)/N$ and

$$d = A_0(\overline{y} - \overline{z})$$

$$= \frac{1}{N}(a_1 y_1 + a_2 y_2 - a_1 z_1 - a_2 z_2)$$

$$= \frac{1}{N}(a_1 y_1 + a_2 y_2 - b)$$

Therefore the coordinates of the center satisfy the following equation:

$$a_1 y_1 + a_2 y_2 - Nd = b$$

Assume that the disc is inscribed to three lines:

$$a_{i1} x_1 + a_{i2} x_2 - b_i = 0, \quad i = 1, 2, 3,$$

and denote the norms of them by N_i, $i = 1, 2, 3$ respectively, then the coordinates of the center satisfy the following linear system of equations:

$$a_{11} y_1 + a_{12} y_2 - N_1 d = b_1$$

$$a_{21} y_1 + a_{22} y_2 - N_2 d = b_2$$

$$a_{31} y_1 + a_{32} y_2 - N_3 d = b_3$$

By our assumption of the gap theorem, we have $|a_{i1}| + |a_{i2}| \leq K, \quad i = 1, 2, 3,$

and therefore

$$N_i = \sqrt{a_{i1}^2 + a_{i2}^2} \leq |a_{i1}| + |a_{i2}| \leq K, i = 1, 2, 3,$$

we can solve the linear equation system for d: $|d| = \overline{D}/D$, where

$$\overline{D} = \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

$$D = \begin{vmatrix} a_{11} & a_{12} & N_1 \\ a_{21} & a_{22} & N_2 \\ a_{31} & a_{32} & N_3 \end{vmatrix}$$

Since $d \neq 0$ and $\overline{D}$ is an integer, we have

$$|\overline{D}| \geq 1$$

Since

$$D = N_1 \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} - N_2 \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + N_3 \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

we have $|D| \leq 3K^3$ and $d \geq 1/3K^3$.

Gap theorem 2. In case that $b_\alpha = 0$ for all α, the shape of any region is a cone. Then in any non-degenerate region we can find an integer point such that the binary length of its coordinates is bounded by $log(K^{n+1}(n+1)+1) = O(n \log K)$.

Proof. We add the following hyperplanes to the system:

$$|x_i| = 1, \quad i = 1, 2, ..., n$$

We inscribe a ball in the region as shown in Figure 2.

The radius d of this ball is at least $\frac{1}{K^{n+1}(n+1)}$. Scale and enlarge it alone the cone to a distance $K^{n+1}(n+1)$ from the origin. Then we obtain a ball of radius 1. It is inside the cone. Since the radius of the ball is 1, we can find an integer point in the ball which is in turn inside the cone. The binary length of any coordinate of the integer point is bounded by

$$\lceil log(K^{n+1}(n+1)+1)\rceil = O(n\,log\,K).$$

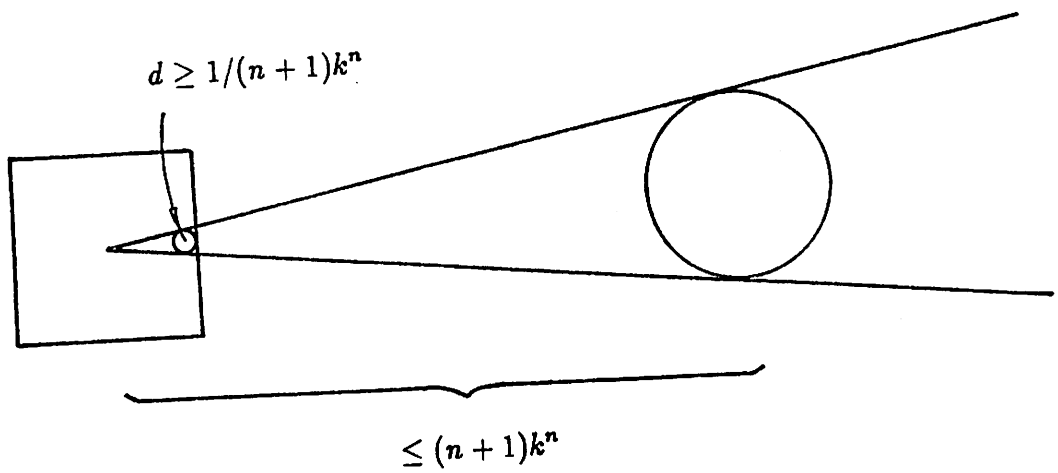

Figure 2.