

Introduction to Neuromorphic Computing

Takashi Inagaki

2019 年 2 月 7 日

目次

| | | |
|-------|-----------------------------|----|
| 第 1 章 | 導入 | 5 |
| 1.1 | Neuromorphic Computing とは何か | 5 |
| 1.2 | この本の対象読者 | 8 |
| 1.3 | この本が取り扱う内容 | 8 |
| 1.4 | ムーアの法則は終わった？ | 9 |
| 1.5 | 消費電力の話 | 9 |
| 第 2 章 | 従来のコンピュータのしくみ | 11 |
| 2.1 | 概要 | 11 |
| 2.2 | MOSFET トランジスタの階層 | 14 |
| 2.3 | 論理回路の階層 | 16 |
| 2.4 | 計算機能 | 18 |
| 2.5 | メモリ機能 | 24 |
| 第 3 章 | 脳の計算法 | 27 |
| 3.1 | 脳の基礎知識 | 27 |
| 3.2 | 脳の情報伝達の仕組み | 28 |
| 3.3 | 活動電位の発生 | 34 |
| 第 4 章 | 脳の数学的モデル | 37 |
| 4.1 | 概要 | 37 |
| 4.2 | ニューロンの数学的モデル | 39 |
| 4.3 | ニューロンの数学的モデル | 40 |
| 4.4 | 脳の学習理論のモデル | 40 |
| 第 5 章 | Neuromorphic Computing | 41 |

| | | |
|-----|---------------------------|----|
| 5.1 | 概要 | 41 |
| 5.2 | アナログかデジタルか | 43 |
| 5.3 | 学習能力をどのように獲得するか | 43 |
| 5.4 | | 43 |

第 1 章

導入

1.1 Neuromorphic Computing とは何か

Neuromorphic Computing とは、大雑把に言えば脳を模倣したコンピュータ，すなわち脳模倣型計算機を作ろうとする研究分野です。なんだ，要するに AI のことかと思われるかもしれませんが。確かに広い意味で言えば AI と呼ばれる領域に入るものではあるものの，現時点（2018）で世間を騒がせている AI とは根本的に異なる点があります。それは，AI を実現しようとするフィールドがハードウェア（物理的）かソフトウェア（仮想的）かという違いです。世間で人気の AI というのはいわゆる機械学習^{*1}と呼ばれるものですが，これはあくまでソフトウェア上で脳の計算法を模倣したものであって，根っこにある物理的なコンピュータの根本的な仕組みは 1940 年代からほとんど変わっていません。これに対し Neuromorphic Computing は，物理的なコンピュータの仕組み自体を脳の機能からインスピレーションを受けて作り直そうというものです。1940 年代から続いている従来のコンピュータをノイマン型コンピュータ，一方で Neuromorphic Computing のような従来と異なる新しいコンピュータのことを非ノイマン型コンピュータ^{*2}といたりします。このノイマン型コンピュータについて知ることは，Neuromorphic 型のコンピュータの利点を知るためにも重要なので，後ほど詳しく触れることにします。

Neuromorphic Computing は，脳の計算方法にインスピレーションを受けて，脳のような特量を持ったコンピュータを作ろうとする取り組みです。脳の計算には特徴的な点がいくつかあります。

*1 機械学習・・・

*2 最も有名な非ノイマン型コンピュータは，ご存知の通り量子コンピュータです。他にも DNA コンピュータといったものがあります。これらのコンピュータは，実現されているかはともかくとしてどれも従来のコンピュータとは全く異なる設計思想で動きます。

- 脳は低消費電力である
- 脳はアナログ計算機である
- 脳はリアルタイム処理を行う
- 脳は汎用ではなく、用途特化の計算機である

1.1.1 脳は低消費電力

圧倒的に重要な特徴の一つとして、脳の消費電力の低さ、エネルギー効率の良さが挙げられます。脳が平均的に消費する熱量はおよそ 20W とされています。この数値は、人間の活動をおよそ 100W としたときに脳のエネルギー消費が約 20% を占めるというところから来ています。^{*3}これに対し、一般的なコンピュータの頭脳（汎用 CPU）の消費電力は、80~100W 程度です^{*4}。もちろん、この数字は単純に比較することはできません。何のタスクに対しての消費電力かが明示されていないからです。脳のエネルギー効率は、並列計算、パターン認識という領域では、従来のコンピュータのそれを圧倒的に上回ります。しかしながら、正確無比な計算が要求される領域ではその限りではありません。従来のコンピュータと脳のエネルギー効率の比較についての詳細は、第2章で論じています。この本では、Neuromorphic Computing の旨味を最大限に活かすべく並列計算、パターン認識という分野を深く掘り下げていきます。

1.1.2 脳はアナログ

脳の計算のもう一つ重要な点として、計算方法がアナログであることが挙げられます。これは、計算をするにあたってデジタル（離散的な、0 と 1）な値を使うのではなくて、アナログ（連続的）な値を用いているということです。従来のコンピュータはすべからずデジタルです。とっっても複雑な 3D ゲームも、Web ブラウザも、スマートフォンのアプリも最終的には 0 と 1 の世界に帰着させることができます。これゆえに、コンピュータは正確無比な計算を行うことができます。一方で、アナログはかなり大雑把な計算をします。脳の神経細胞は、電流とイオンの放出によって細胞同士の情報伝達を図っていますが、その仕組みにはかなりの冗長さがあります。多少のエラーも許容できるというシステムがアナログ計算の特徴といえます。多くの研究者は、その間違いを起こすシステムにこそ本質がある

*3 (要出典!!!!!!!!!!)

*4 (データ持ってこよう)

のだと考えているようです。従来のコンピュータの計算方法については第二章で、脳の情報伝達の仕組みについては第三章で扱います。

1.1.3 脳はリアルタイム処理

さらに、もう一つの特徴として、脳はリアルタイムで処理を行うということが挙げられます。これは、脳というデバイスが環境からの光や音などの入力情報を受けて即座に何らかの出力を返すということです。我々が走っている車を見て、これは車だ、あれはタクシーだ、などと瞬時に見分けることができているのは、我々の脳が視覚からの情報をリアルタイムで処理しているからに他なりません。このリアルタイム処理というのは従来のコンピュータにとってはなかなか難しい仕事です。例えば、複雑な自然環境で自立歩行するロボットを考えてみましょう。外部からの情報を入力として受けてロボットが安全に行動するためにはリアルタイムな処理が求められますが、それに答えるための高性能な計算機を積もうと思うと、電力的にも重量的にも非常に厳しいものがあります。そもそもある種の問題はどんなに高性能なパソコンを積んでもリアルタイム処理が難しいということもあるかもしれません。なぜ従来のコンピュータにリアルタイム処理が難しいのか、それは計算がデジタルであるというところに依存しています。従来のコンピュータは、外部の光や音などの情報を一旦アナログからデジタルに変換して、0と1の離散的な値の計算で物理的な現象を、人工的に作った複雑な機構で計算、シミュレートしてからその結果を予測します。一方で、脳は外部の情報をアナログのまま入力し、脳自体を物理現象のエミュレータとして用いることで高速に計算を行なっているのです。詳しい話は、コンピュータと脳のしくみについて学んでからすることにします。

1.1.4 脳は汎用ではなく、用途特化の計算機

最後の特徴として、脳は汎用計算機ではなく、ドメイン特化な、用途特化な計算機であるということが挙げられます。従来のコンピュータは汎用計算機ですが、脳は汎用ではありません。特殊な事柄を計算するために特化したデバイスです。これはどういうことかというと、例えば物体認識や音声認識、食感認識や味覚認識、嗅覚認識といった特定のそれぞれの用途を計算するための機構がよりどりみどり詰まっている計算機が、脳であるということです。なぜなら、脳は進化の過程で、自然選択を受けて長い時間をかけて生み出された自然の産物であるからです。

1.2 この本の対象読者

この本は、脳について興味がある人すべてが対象読者です。脳について直接の興味がなくとも、より効率的なコンピュータについて学びたい人にとっても有用でしょう。コンピュータについてすでに専門的な知識を持っている場合、この本の第二章は飛ばして読んでも差し支えありません。また、脳についてすでに深い理解がある場合は第三章から第四章は読み飛ばしてもらって構いません。この本は学際的な領域かつ、そこにおける基礎的な話題に触れるに止まっているので、より専門的な話題を知りたい方は、各分野の専門的な教科書なり論文なりを読むことをお勧めします。例えば、コンピュータについてより詳しく知りたい場合、コンピュータの構成と設計などはいい教科書でしょうし、脳について詳しく学びたい場合はカandel神経科学などがいい選択でしょう。

読者層は、かなり広く想定しており、コンピュータや脳についての知識がほとんどない場合にも問題なく読めるように書いているので、専門家はやや退屈すると思います。ガチの専門家は本書の対象外であるとも言えるかもしれません。もちろん、内容を吟味してくださるのは大歓迎です。基本的には、何も知らずとも読めるようにしています。基本的な物理の知識、（とはいっても高校程度の内容）は予備知識として想定しています。

1.3 この本が取り扱う内容

この本は、Neuromorphic な計算機とはどのようなものであるか、というところを従来のノイマン型コンピュータと比較して議論している本です。そのためのコンピュータのしくみ、脳のしくみについての基礎知識を第二章から第五章の間で提供しています。そもそもコンピュータって何なのか？脳って何なのか？何をしているのか？そういった疑問にもある程度答えようと頑張っています。まず、コンピュータの仕組みを電子レベルから見ていった上で、どのようにしてコンピュータは計算を行なっているのかという根本原理を学びます。次に、脳の基本的な知識をおさらいした上で、脳を計算機として見たときにどのように脳を捉えることができるかについて見ていきます。最後に、脳を模した計算機を作るためにはどうすればよいのか？について見ていきます。

まず、第二章で従来のノイマン型コンピュータの仕組みについて、電子のレベル

から回路デザインのレベルまで一通りさらいます。この章で、従来のコンピュータがどのようなものであるかが理解できるはずです。

1.3.1 本書の狙い

1.4 ムーアの法則は終わった？

近年の Neuromorphic Computing への注目の高まりの要因の一つとして、ムーアの法則の終焉があります。ムーアの法則というのは、1965年にゴードンムーアが提唱した説で、18ヶ月ごとに集積回路の素子数が2倍になるというものです^{*5}。この予測はごく最近まで実によく的中していました。ゴードンムーアの予測が正確過ぎたのか、予測に的中させるためにみんなが頑張り過ぎたのかは分かりませんが（両方あると思います）、実際に集積回路に詰め込めるトランジスタの数は指数関数的に増加してきました。しかし、ここ数年その流れに陰りが見えています。というより、ここ数年においては完全に素子数の増加がストップしたと言ってよいでしょう。実際、2017年5月に大手半導体メーカー NVIDIA の CEO が、ムーアの法則は終わったと公式に発言しています^{*6}。その理由はいくつか挙げられます。一つは物理的な大きさの限界です。現在の時点で一つのトランジスタの幅は数十 nm 程度の大きさしかありません。これはおよそ原子数百個分の大きさです。原子より小さい構造を作ることは現実的ではないので、最後に立ちはだかる大きな壁としてこのトランジスタサイズの限界問題があります。実際には、この段階にいくまでに別の問題に阻まれます。それが漏れ電流と呼ばれる電子の漏れにより生じる、消費電力の増加、さらには発生する熱量の増加です。コンピュータは、電子の流れ（電流）を操作して演算を行なっているので、電流にはとても敏感にならなければいけません。トランジスタのサイズを小さくすればするほど、この漏れ電流の影響が大きくなります。漏れ電流についての詳細は量子論的な話が絡んでくるのでここでは扱いません。

グラフ

1.5 消費電力の話

Neuromorphic なコンピュータは、もしも上手く実現することができれば、パターン認識のような分野において圧倒的に計算にかかるエネルギー効率が良くな

^{*5} GR Moore(1965). "Cramming More Components onto Integrated Circuits"

^{*6}

ると考えられています。従来のコンピュータと比べて 1000 倍のオーダーで計算できると示唆する研究結果もあります*7。その根拠となるのが、我々の脳の計算の仕組みです。例えば、我々が視覚から物体を認識するプロセスを考えてみましょう。

我々の脳が物体を見て、それが何かを無意識的に認識するプロセスは、以下のように行われます。まず、眼の網膜における視細胞が外部からの光に反応して電気信号を脳の視床と呼ばれる部分に送ります。この段階では、どこが明るくてどこが暗いかといったコントラストの情報や、どこがまっすぐでどこが曲がっているかといったエッジの情報しか分析されていません。次に、その情報が視床から脳の後頭野にある一次視覚野と呼ばれる部位に送られて、その過程でより高等な情報、物体の立体的な配置などが認識されます。その後、さらに情報は下側側頭野に送られて、そこで物質の名前を確定的に特定するまで情報が処理されます。この脳の構造は、計算をするために編み出されたコンピュータのしくみとは全く違い、物体認識というタスクを行うためだけに高度に最適化されたハードウェアといえます。

図

機械学習における物体認識は、このハードウェアをモデル化して、コンピュータの上で大量の計算をして行なっています。もうお分かりかと思いますが、このやり方は物体認識に最適化されたものではありません。コンピュータの万能性を利用してなんとか物体認識を計算上で行なっているのです。一方で、我々の脳みその一部ははじめから物体認識をするためだけに、厳しい自然選択を勝ち抜いて来た非常に高度なハードウェアです。このような理由から、物体認識において、効率の面で脳の方が圧倒的によいのは明らかです。

*7 どここの研究結果だ？

第 2 章

従来のコンピュータのしくみ

2.1 概要

コンピュータのしくみを学ぶのはなかなか厄介な仕事です。というのも、パソコンやスマートフォン、IoT などとさまざまなコンピュータがありますが、それらのすべてには最先端の技術が詰まっており、学び方の切り口によって何冊もの分厚い教科書が必要となってくるからです。例えば、コンピュータを作り上げるハードウェア、(コンピュータの物理的な構成)のしくみを詳しく学ぼうとするならば、大学で学ぶ物理学、電子回路、離散数学などの知識は必須です。ソフトウェア(ハードウェアをどのように操るか)について学ぼうとすれば、書店のパソコン関連の棚を思い浮かべれば分かるようにありったけの本があります(書店に並んでいるほとんどのコンピュータ関連の本は、ソフトウェアについてです)。例えば、Youtube はどうやって動いているの?とか、Web ページはどうやって動いているの?といった質問に詳しく正確に答えようとするれば、とんでもない知識量が必要になってきます。

ここでは Neuromorphic Computing との比較のために、基本的なコンピュータの計算機としての構成を学ぶという目標を掲げます。そのような目的のために学ぶべきは、根本となるハードウェアの実装の部分です。そのため、この章ではコンピュータを解体して、結局コンピュータは物理的になにをやっているの?というところに答えていくことにします。

一般的なデスクトップコンピュータを図 1 に示します。このようなコンピュータは、CPU、メインメモリ、電源、主記憶装置(ハードディスク)などから構成されています。この章で注目するのは CPU とメインメモリです。その他の機器もも

もちろん無くてはならないものですが、コンピュータの計算のしくみを探る上ではそれほど重要ではありません。むしろ、コンピュータの本体はこの CPU とメインメモリだと言えます。CPU は Central Processing Unit の略称で、中央演算処理装置と言います。文字通り CPU はコンピュータの頭脳ですが、CPU だけではまともに動くことができません。もちろん計算することはできるでしょうが、計算するたびに以前の結果を忘れてしまう機械なんてのは、なかなか使いこなすのが難しそうです。そもそも、人間が忘れっぽくて単純な計算が苦手だからコンピュータを作ろうとしたのに、それでは我々と大差なさそうです。そこで必要になるのが計算した、もしくは計算したいデータを保存する機器、メインメモリです。CPU はメインメモリにあるデータを取って来てはじめて、マシンな計算をすることができます。言うなれば、人間が計算をするときに紙とペンがなければまともなかけ算もできないようなものです。人間との大きな違いは、計算の処理の仕方、計算速度、そして正確さです。この章では、コンピュータの計算の処理の仕方を主に学ぶことになります。メインメモリは大規模なデータの保存地点です。CPU とメインメモリとが連絡を取り合って、データをやり取りすることで正確無比なコンピュータとしての真価を発揮するというわけです。

コンピュータの計算の仕方を理解するにはひとまず CPU とメインメモリに注意すれば良いことを伝えました。次に、どのような順番でコンピュータを学んでいくのかについて記します。まず、CPU とメインメモリはどちらもナノレベルの電子回路から成り立っているというところで共通しています。しかしながら、CPU の回路の構成とメインメモリの回路の構成は大きく異なるばかりか、CPU の構成の方がかなり複雑です。この理由から、以下の章では CPU の構成をまずはじめに扱います。その後メインメモリについて扱きましょう。

CPU は 10 億個以上のトランジスタ（電気的なスイッチ）からなる大規模な回路の集まりです。それぞれのトランジスタが簡単な回路を作り、簡単な回路が複雑な回路を作り、複雑な回路が超複雑な回路を作り、といった手順で CPU が構成されていきます。あまりに複雑なので、CPU を理解するためには抽象化の階段を一步ずつ降りていく、もしくは上がっていくことが必要になります。この本では、CPU を電子のレベルから目に見えるチップのレベルまで、下から上に一步ずつ登っていくことを目指します。CPU の一番最下層は電子を操作する領域、すなわち電子回路です。そこには、MOSFET トランジスタと呼ばれる極小の電気的なスイッチがあります。トランジスタを電流が流れる時、つまりスイッチが ON に

図 2.1 CPU とメモリ

なるときを 1, 電流が流れていない時を 0 と勝手に決めることで, 物理現象である電子の流れ, 電流をアナログ的な見方からデジタルへと昇華させます。ここがアナログとデジタルの分岐点です。このようにして電流を 2 進数のデジタル情報として捉えることができれば, あとはこれらのスイッチを数学的に上手く動作するようにはめ込んでいくだけです。もちろん, この「だけ」の作業が恐ろしく複雑であるのは言うまでもありません。

MOSFET トランジスタによるスイッチの次の階層は, これらのスイッチを組み合わせて作る論理回路です。例えば, トランジスタを複数組み合わせることで数学における AND, OR, NOR といった論理回路を組み上げることができます。AND 回路や OR 回路をパズルのように組み合わせることで, 足し算を行う回路や, 引き算を行う回路, 二つの数字の大小を比べる回路など, さまざまな論理回路を構成することができます。こうなるといろいろなものを作ることができ, だんだん楽しくなって来ますね。ここで一つ注意することがあります。さきほど, CPU はメイ

ンメモリがなければなにも記憶できないと言いましたが、実際には CPU 内部にもデータを記憶する機構が存在します。それは、レジスタや、キャッシュと呼ばれる記憶機構です。^{*1} CPU 内部のレジスタ、キャッシュはメインメモリとは異なる仕組みで動作しています。この点についても、後ほどこの章で扱います。

MOSFET トランジスタによるスイッチの階層、これらのスイッチを組み合わせる論理回路の階層、そして最後に、マイクロアーキテクチャと呼ばれる階層を見ていきます。マイクロアーキテクチャとは、これらの論理回路をどのように配置し、どのように計算させるか、という最終的な CPU の設計図となるものです。マイクロアーキテクチャを理解することができれば、その CPU の物理的な構成はほとんど把握したと言ってもよいでしょう。例えば、足し算だけ行わせるような CPU にしたければ、加算器を置きまくるだけにしてみるだとか、ベクトル演算に特化させたいければ、ベクトル演算用のユニットをたくさん配置するというような工夫ができます。実際にどのようにして計算させるのかについて把握するには、そのマイクロアーキテクチャに対応する計算の仕方、データの取り扱い方なども決める必要があります。また、電子は回路を無限の速さで流れるわけではないので、回路の遅延の影響についても考える必要があるでしょう。

2.2 MOSFET トランジスタの階層

MOSFET は、現代のコンピュータを作るために不可欠な、小さな電氣的なスイッチです。先にも説明したように、MOSFET が CPU のすべての土台となっています。この節では、MOSFET とは一体なんなのか、MOSFET はどのような仕組みでスイッチとしての役割を果たすのか、MOSFET はどのような電氣的性質を持つのかという3つのポイントをおさえます。この3つのポイントを抑えることができれば、MOSFET が何なのか、MOSFET がコンピュータにおいてどのような役割を演じているのかという疑問に答えることができるはずです。

2.2.1 MOSFET とはなにか

MOSFET は Metal Oxide Semiconductor Field Effect Transistor の略称で、日本語では金属参加 h a n n d o u t a i 電界効果トランジスタといいます。

^{*1} 実はこの、どこにデータを保存するのかという問題が計算を行う上で一番重大な問題になっています。計算をするためにはデータを取ってこなければならないというのは前に伝えた通りですが、現在のコンピュータの仕組みでは一般的に計算にかかる時間よりもデータを輸送するのにかかる時間の方が長くなってしまっているからです！ (((要出典)))

図 2.2 全体像

MOSFET には大きく分けて pMOS と nMOS の二つのタイプがあり，それらを組み合わせた cMOS(complementary MOS) というものがコンピュータにおける論理回路の基本素子として使われています。ここでは，段階を踏んで nMO と pMOS，そして cMOS へと進み，cMOS がどのようにして回路に組み合わされていくのかについて見ていきます。

図も入れておこう。(基本的な MOS について触れるのはもちろんだが，最新の MOS テクノロジーについても触れたいというか知りたい。3D のとか)

2.2.2 pMOS と nMOS

MOSFET は，pMOS と nMOS の 2 種類に大別されます。その違いを探るために，まず簡単に半導体についておさらいしておきます。世の中の物質は，電気の通しやすさという観点から三つの種類に分けることができます。それぞれ電気の通しやすさの順番で，導体・半導体・絶縁体です。MOSFET は半導体特有の性質

を用いて、スイッチとしての機能を実現しています。半導体には、n型半導体とp型半導体の2種類があります。ある物質が電気を通しやすいかどうかというのは、物質中の電子がどれだけ動きやすいかということに依存します。例えば絶縁体の場合、物質を構成する原子の中には自由に動ける電子、または正孔はありません。金属のような導体の場合、高い密度で原子が互いに接近し合っているため、電子軌道を共有し合っており、電子は自由に動き回ることができます。このような理由から、絶縁体は電気を通しにくかったり、導体は電気を通しやすかったりするわけです。一方半導体は、

2.3 論理回路の階層

2.3.1 論理ゲートの作成

cMOSを複数用いることで、論理ゲートという論理回路の基本となる回路を組み立てます。その論理ゲートを複数組み合わせることで、コンピュータは複雑な機能を実装しています。代表的な論理ゲートにはANDゲート、ORゲート、NANDゲートなどがあります。実際にCMOSを用いたそれらの論理ゲートの物理的な回路の構成を見る前に、真理値表という便利な表を紹介します。言葉で説明するよりも図で見てもらった方が早いので下に図を用意しました。

真理値表は、入力された値に対応する結果を表した表のことです。ANDゲートの場合、入力されたすべての値が1の場合のみ、結果に1が帰ってきます。それ以外の入力のパターンではすべて0が帰ってきます。ORゲートの場合、すべての入力のうち少なくとも一つが1であれば、結果に1が帰ってきて、入力が全て0の場合のみに0が帰ってくるという関係になっています。この真理値表を用いることで、論理ゲートのすべてのパターンにおける入出力の振る舞いを把握することができます。

では、実際にcMOSを用いてNANDゲートとANDゲートを作ってみましょう。読者の皆さんは、余裕があればまず自分で考えてみてください。しばらく考えたあと、次ページの回路図を見てください。

実際にこれは真理値表に対応する機能を果たしてくれるのでしょうか？入力が1のラインと0のラインを色分けながら別の紙に書いたりして、確かめてみてください。ちゃんと入力がどちらも1の時のみに出力が1になることが確認できると

図 2.3 <https://introcs.cs.princeton.edu/java/71boolean/>

思います。

さて、このようにして cMOS を用いて論理ゲートを作成することができました。それはいいものの、論理ゲートを登場させるたびにいちいち cMOS のレベルで考えるのはとても大変です。そこで、今後はこれらの論理ゲートを簡略化させた記号を用いることにしましょう。この抽象化の考え方は、複雑なコンピュータのしくみを探る上でとても重要です。なぜ重要かという点、人間の脳みそは論理ゲートの機能を考えている時に同時に MOSFET レベルでの動作を考慮できるほど優れていないからです。これはコンピュータに限らず、複雑なシステムを相手にする時に非常に有効な手段になるでしょう。簡略化した記号を下の図に示します。

以降は、これらの記号を用いてさらに複雑な回路を作っていきます。まだまだ論

図 2.4 <https://electronics.stackexchange.com/questions/226023/cmos-and-gate-implementation>

理ゲートのレベルでは CPU を作ることはかないません。もっと複雑な回路を見ていくことにしましょう。

2.4 計算機能

まず、この図を見てください。

これは、MIPS アーキテクチャというコンピュータの構成の一つで、コンピュータの仕組みを学ぶ題材としてよく用いられます。もちろん、実際によく使われているコンピュータアーキテクチャの一つでもあります。今後しばらくの間の目標は、この回路図に書かれているパーツがどのようなものなのか、一つ一つ理解していくことです。それらのパーツを理解することによって、コンピュータがどのように動いているのかという大目標に近づいていきます。

この節では、先ほど作った論理ゲートを用いて、足し算を行う加算器と呼ばれる回路、いくつかの入力のうちから一つを選択して出力するマルチプレクサと呼ばれる回路などの、より複雑な機能を持つ回路を見ていきます。

図 2.5 <https://www.conceptdraw.com/How-To-Guide/logic-gate-diagram>

2.4.1 加算器

加算器は、入力された値同士を足し算して出力する回路です。最も簡単な加算器は 2 つの値を入力として受け付けて、その値同士を足し、一つの答えと一つの桁上がり情報を出力する 2bit の足し算です。このような、桁上がり情報を出力するのみで入力を受け付けない足し算回路のことを、半加算器 (Half Adder) と呼びます。桁上がり情報というのは、足し算の結果桁が上がるのならば 1 を、上がらなければ 0 を出力するものです。ここで扱う数字はすべて 2 進数ですから、例えば入力が 0 と 1 ならば答えは 1 で、桁上がり情報は 0 となります。もし入力が 1 と 1 ならば、答えは 0 で、桁上がり情報は 1 になるというわけです。

一方で、2 つの計算する値の入力以外に、他の足し算回路からの桁上がり情報も入力として持つような足し算回路のことを全加算器 (Full Adder) と呼びます。もしも、二つの入力値が 1 と 0 で、桁上がり情報の入力が 1 だった場合、全加算器は答えの出力として 0 を、桁上がり情報の出力として 1 を返すということになります。

半加算器と全加算器は、論理ゲートを用いてそれぞれ次のように表すことができ

図 2.6 MIPS アーキテクチャ

図 2.7 半加算器

ます。

お気付きのように，ここまでの加算器では2つの数字を足すことしかできず，もっと大きな数同士を足したい時には困ってしまいます。しかし，大きな桁の足し算はこれらの半加算器，全加算器を複数組み合わせることで簡単に実現できます。計算速度を気にするのならば，かなり複雑な仕組みが必要になってくるので簡単とは言えなくなってきますが，そこを厭わなければ単純に繋げるだけで大きな桁の足し算が実行できます。そのような加算器を順次桁上げ加算器（ripple-carry adder）と呼びます。

図 2.8 全加算器

図 2.9 半加算器回路

同様にして，引き算を行う回路，かけ算を行う回路，割り算を行う回路を作ることができます。本書ではあえて扱いません。以下の記号で省略させてもらうことにします。詳しく知りたい方は，このページの脚注^{*2}にあるような本で学んでみてください。

*2

図 2.10 全加算器回路

図 2.11 順次桁上げ加算器

2.4.2 マルチプレクサ, デコーダ

この節では、四則演算のような計算をする回路だけでなく、入力された複数の値から一つを選択するような回路や、入力された値のパターンから複数の値を出力するような回路などを見ていきます。はやくコンピュータを作れよ、という声が聞こえてきますが、これらは全てコンピュータを構成するために必要なパーツです。

マルチプレクサ (Multiplexer) は、入力された値のうちから、制御信号の値に応じて一つを選択し、出力する回路です。マルチプレクサは、MUX と略して表記されることが多いです。例えば、入力が二つの MUX ならば、必要となる制御信号は一つです。なぜなら、制御信号一つ (0or1) で二つの値を選択することができるからです。

図 2.12 2:1MUX

入力が N 個の MUX の場合、必要となる制御信号は $\log_2 N$ 個となります。

図 2.13 N :1MUX

デコーダ (Decoder) は、入力された値の組み合わせによって、複数ある出力のうちから一つを選び出力する回路です。例えば、出力が 4 つあるようなデコーダならば、そのうちのどの一つを有効にするかを選ぶのに 2 つの入力が必要になります。出力が 16 つならば入力が 4 つとなります。MUX でいう制御信号の数と対応していますね。

入力が N 個の場合、出力のパターンは 2^N 個ということになります。

図 2.14 2:4 Decoder

2.5 メモリ機能

これまで扱ってきた回路は、入力された値に対して即座に出力を返すような回路でした。つまり、値を保存するという概念がなかったのです。しかし、実際のコンピュータは値を一時的に保存することによって複雑な計算を実行しています。逆に言うと、値を保存することができないコンピュータは大したことないモノになってしまうということです。

回路に記憶機能を持たせることは、時間の概念を導入することに他なりません。というのも、値を保存するならば、その値をいつ入力するのか、いつ読み出すのか、いつ書き換えるのかという時間の流れが登場してくるからです。このように、時間の流れを取り入れた回路を順序回路 (Sequential Logic) と呼びます。一方で、以前までに扱った時間の流れを考えずその場その場で処理を行うような回路のことを組み合わせ回路 (Combinational Logic) と呼びます。

この節では、そのように値を保存してくれるような回路を考えます。具体的には、ラッチ回路、フリップフロップ回路と呼ばれるものです。これらを組み合わせることで、いわゆるレジスタというやつが作れます。

2.5.1 ラッチ回路

2.5.2 フリップフロップ回路

第3章

脳の計算法

これまでの章で、従来のコンピュータがどのようにして計算を行なっているのかについて扱って来ました。はじめに言ったように、Neuromorphic Computing は脳の計算法からインスピレーションを受けています。よって、我々の脳がどのようにして計算を行なっているのかについて学ぶことも不可欠です。この章では、脳の基本的な構成要素を解説したのちに、数ある脳の処理の中でも特にヒトの物体認識メカニズムに焦点を当てて、脳がどのように計算を行なっているのかについておおまかに見ていきます。その後、脳の計算法の様々な数学的モデルを紹介します。

ところで、Neuromorphic Computing を考える上でなぜ脳の数学的モデルを取り扱う必要があるのでしょうか？その答えは、仕組みの分からないものは作ることができないからです。

3.1 脳の基礎知識

我々の脳は、外から見ると大きく大脳皮質 (cerebral cortex), 脳幹 (brain stem), 小脳 (cerebellum) の3つに分けることができます。大脳皮質は領域によってさらに4つに区分することができ、それぞれ前頭葉, 頭頂葉, 側頭葉, 後頭葉と言います。脳幹は中脳, 橋, 延髄をまとめた呼称です。小脳は小脳です。

いま、脳の各領域に名前をつけてそれぞれ区別しましたが、すべての領域の構成単位は共通しています。脳を構成する基本単位は、ニューロン (neuron) とグリア細胞 (glial cell) です。これは、恐ろしく複雑なコンピュータが結局のところ数十億の MOSFET トランジスタで構成されていることに似ています。しかしながら、神経細胞やグリア細胞のバリエーションは、MOSFET に比べて格段に柔軟で幅が広いです。そのバリエーションの多さに加えて、単純な数だけでも神経細胞の数は

図 3.1 脳の領域 (Wikipedia より)

一千億ほどあると言われてしています*1。これから、神経細胞やグリア細胞の性質とその多様性に触れていきます。

典型的なニューロンは、中心に細胞体 (soma) を持ち、そこからいくつかの樹状突起 (dendrite) と一本の軸索 (axon) を伸ばしています。通常、ニューロンは他のニューロンからのシグナルを樹状突起で受け取り、細胞体で情報を統合したのちに、軸索で他のニューロンに情報を伝えます。その情報伝達のしくみは、主に電気信号と化学物質の伝達によっています。

これはあくまでも典型的なニューロンの図であることに注意してください。先ほど述べたように、神経細胞は非常に多様性にあふれています。例えば、細胞体からの突起の数や、その広がりによって非常に多くの種類のニューロンが生まれます。次の図 3.3 はニューロンの多様性をよく表しています。

3.2 脳の情報伝達の仕組み

この多様性を踏まえた上で、ニューロン同士がどのようにして情報をやり取りするのかについて見ていきます。端的に説明すると、ニューロンは外部からの入力を電気信号として受け取り、その積算がある閾値を越えると自身が電位を大きく変化

*1

図 3.2 典型的な神経細胞

させて、軸索から出力先のニューロンへと電流を伝えます。その時に発生する電位のことを活動電位（action potential）と呼びます。ニューロンの情報伝達の仕組みを理解するには、1. ニューロンはどのようにして外部からの入力を受け付けているのか 2. 受けた入力は細胞体の中でどのように処理されるか 3. 他のニューロンにどのようにして出力を行うのかの3点を理解する必要があります。

3.2.1 ニューロンはどのようにして外部からの入力を受け付けているのか

さきほど見たように、他のニューロンからの入力を受けるところは樹状突起と呼ばれる部位で、外部へと出力をするところは軸索でした。つまり、ニューロン間で情報伝達が行われるのは、受け手の樹状突起と送り手の軸索が繋がっている領域です。この領域のことをシナプス（synapse）と呼びます。シナプスには、送り手からの電気信号をそのまま受け手に伝える電気シナプス（electrical synapse）と、送り手からの電気信号を一旦化学物質に変換して受け手に伝える化学シナプス（chemical synapse）の2種類があります。また、送り手自身のことをシナプス前細胞、受け手自身のことをシナプス後細胞と呼びます。

化学シナプスでは、活動電位が前細胞の軸索の終末に達すると、それに反応して化学物質が軸索終末と樹状突起の先端の間のわずかな隙間に放出されます。この

図 3.3 (<https://www.janelia.org/lab/spruston-lab/research/cellular-diversity-in-the-hippocampus>)

化学物質が後細胞の樹状突起表面にあるイオンチャネルと結合すると、イオンチャネルが開閉します。イオンチャネルを通じて細胞膜内外のイオンの分布が変化することによって、後細胞の電位が結果的に変化することになります。このような仕組みで、前細胞の電気信号が間接的に後細胞に伝わります。また、この隙間のことをシナプス間隙 (synaptic cleft) と呼び、その感覚は 20-40nm ほどです。

通常、この化学シナプスにおける遅延は数ミリ秒^{*2}に及びます。この数字は、コンピュータと比べると格段に遅いことが分かります。現代の CPU は数 GHz で動作するので、1 サイクルにかかる時間は数百ピコ秒から数ナノ秒です。ただし、これはあくまで化学シナプスにおける前細胞と後細胞のやり取りにかかる時間で

*2

図 3.4 シナプス (Khan Academy)

あって、脳とコンピュータの計算方式は大きく違うので、単純な比較はできないことに注意してください。

また、このような仕組みから、化学シナプスは非常に多様な情報の伝達構造を持っていることが分かります。化学シナプスは、活動電位の大きさ、化学物質の量、化学物質の種類、シナプス間隙の幅、受け手のイオンチャネルの数、イオンチャネルの活性度合い、電位の発生度合いなど様々な要因が絡み合っています。この多様性はコンピュータにおける情報伝達とはまったく異なる大きなポイントの一つです。

電気シナプスでは、シナプス間隙の間に直接前細胞と後細胞を繋ぐ連絡路のようなものがあり、活動電位を直接伝えます。

図 3.5 化学シナプスと電気シナプス (<https://www.nature.com/articles/nrn3708>)

この化学シナプスと電気シナプスの割合は、

3.2.2 受けた入力細胞体でどのように処理されるか

外部のニューロンから受けた入力、その後どのように統合されていくのかについて見ていきます。具体的には、活動電位の発生のメカニズムです。樹状突起から電気信号の入力を受けたニューロンは、その入力の大きさやパターンによって活動電位の発生の様子が変わります。また、活動電位の発生は、ニューロンの細胞膜常に存在するイオンチャネルの数や分布、外部の組織液の様子にも依存しています。

そもそも、活動電位とはニューロンの細胞膜の内部と外部における電位の差の急激な変化のことです。活動電位のない通常の状態では、ニューロンとその外

部の組織液の間に約 65mV の電位差があります。慣例として、細胞膜の外側の電位を基準の 0V として扱うので、安定状態のニューロン内部の電位は-65mV ということになります。この-65mV の電位のことを、静止膜電位 (resting membrane potential) と呼びます。静止膜電位が外部のニューロンからの入力などによって変動しある閾値を越えると、急激に電位が変化し活動電位が生成されます。急激な変化というのがポイントになります。活動電位が発生する閾値まで電位が達しなかった場合、それまでに入力された情報は次のニューロンへと伝達されません。これを全か無かの法則と言ったりします。

図 3.6 <http://n.neurology.org/content/82/11/989>

すべての膜電位の変動は、ニューロンの細胞膜上にあるイオンチャネルを介して、細胞膜外のイオンと膜内のイオンが移動することによって発生します。その際に重要な役割を果たしているイオンは、 Na^+ 、 Ca^{2+} 、 K^+ 、 Cl^- の 4 つです。この主要な 4 つのイオンが安定的に存在している状態の電位が -65mV というわけです。ニューロンの細胞膜は、脂質の二重膜で覆われており、ところどころにイオンチャネルが刺さっています。脂質はイオンを通さないのので、イオンチャネルを通してのみイオンが移動することができます。このイオンの分布が電位を生み出しているのです。

イオンの移動は、イオンの濃度勾配（濃い方から薄い方へ流れる）の力と、電気勾配（クーロン力による引力）によって決まります。

図 3.7 neuron ion concentrations

3.3 活動電位の発生

活動電位は、神経細胞の細胞内と細胞外の電位差によって生じます。活動電位発生の元となる電位差は、細胞膜上にある特定のイオンを通過させたり移動させたりすることのできるイオンチャネルやイオンポンプなどによって、細胞内外のイオンが濃度勾配的・電氣的に移動、もしくは能動的に輸送されることによって生じます。細胞膜は脂質で覆われており電氣的な遮断性を持っています。イオンチャネルやイオンポンプなどといったイオンの通路によってイオンの移動が起こり、細胞膜間に電位差が生じ、それがある程度連続して起こることによって閾値を超えると活動電位が発生して電流が流れ、次の神経細胞へと情報が伝達されていきます。通常時、つまり活動電位を発生していない時の細胞膜の電位は-65mV になっています。これを静止電位と呼びます。

イメージ図

活動電位を生成するにあたって重要な役割を果たしているイオンは主に 3 つ挙

げられます。ナトリウムイオン，カリウムイオン，そしてカルシウムイオンです。細胞膜が静止状態にあるとき，ナトリウムイオンは細胞内には～，細胞外には～，カリウムイオンは細胞内には～，細胞外には～あり，カルシウムイオンは細胞内に～，細胞外に～あります。

| イオン | 細胞内濃度 (mmol) | 細胞外濃度 (mmol) |
|-----------------|--------------|--------------|
| Na ⁺ | 50 | 440 |
| K ⁺ | 400 | 20 |
| Cl ⁻ | 52 | 560 |

絵も描いとく

このような細胞な以外における濃度の差によって濃度勾配が生じ，イオンチャネルを通してイオンの流れが発生します。イオンの分布が変化することによって細胞膜にかかる電圧が変化するのです。ここではまず，ナトリウムイオンの動きから見ていきます。ナトリウムイオンは図のように細胞～の方が多くっており，ネルンストの式から細胞膜間に発生する電位を計算すると，ナトリウムイオンの濃度が細胞内外で釣り合うような電位が計算できます。

$$V_{Na^+} = \frac{RT}{zF} \ln\left(\frac{[Na^+]_{内}}{[Na^+]_{外}}\right) \quad (3.1)$$

R は気体定数，T は温度，z はイオンの価数，F はファラデー定数で，Na_内⁺は細胞内のナトリウムイオンの濃度，Na_外⁺は細胞外のナトリウムイオンの濃度です。これに室温 25℃とそれぞれの定数，そして表の濃度を代入すると，細胞内外のナトリウムイオンが釣り合う電位（平衡電位，ネルンスト電位）はおよそ +55mV と計算できます。同じようにしてカリウムイオン，塩化物イオンの平衡電位を計算すると，以下の表のようになります。

| イオン | 細胞内濃度 (mmol) | 細胞外濃度 (mmol) | 平衡電位 (mV) |
|-----------------|--------------|--------------|-----------|
| Na ⁺ | 50 | 440 | +55 |
| K ⁺ | 400 | 20 | -75 |
| Cl ⁻ | 52 | 560 | -60 |

細胞膜の静止電位が-65mVであるのに比べると，Na⁺の平衡電位が高いように思えますが，Na⁺を通すイオンチャネルは静止時にはあまり開いていないためうま

く釣り合っています。なんらかの要因^{*3}によって、このイオンのバランスが崩れることによって細胞膜の電位が少し上昇すると、閉じていた Na^+ のイオンチャネルがだんだん開き始めます。この変化が連続的に起こることで、急激に細胞膜の電位が上昇し、活動電位が発生するのです。

^{*3} 他の神経細胞からの電気信号の入力など。

第4章

脳の数学的モデル

4.1 概要

これまで、脳が物理的にどのように動作しているのかについて大雑把に見てきました。この章では、そんな脳の働きを数学的にどのように解釈できるかというところについて見ていきます。脳を数学的に理解するためには、いくつかのアプローチがあります。一つ目は単純に脳の物理的な特性を忠実に再現するモデル、二つ目が脳の機能から本質的な特性を抜き出し、作り上げたモデルです。その取り組み方はさらにミクロかマクロかという視点で分けられます。一つの神経細胞の物理的な特性を忠実に再現したモデルは、確かに神経細胞の特徴をよく分析することができるかもしれませんが、情報処理の観点から言えば少し具体的すぎ、かつ計算的に高価かもしれません。脳の情報処理は、多数の神経細胞が繋がりあっているというところに本質があるので、一つ一つよりも集団としての神経細胞の振る舞いに注目した方がより本質を抜き取れそうです。

脳の計算モデルを考えるときには、どの要素がどこまで計算処理に関わっているのかというところについて考えなければなりません。ニューロンの細胞膜上に存在するすべてのイオンチャネルの動作を含めたモデルは、計算的にとても高価になりますし、なによりミクロなすべての分子の運動を追い切ることなどとても不可能です。そもそも、計算能力を問題にするならば、ニューロンを物理的に完璧に再現したモデルを作ればよいという考え方自体が根本から間違っている可能性があります。ニューロン自身にも計算にあまり関係ない要素が（自分が生きるための機構など）あるはずで、一般に、ヒトの脳は1000億^{*1}ほどの神経細胞が存在すると言われています。この膨大な数の相互作用の中に計算の本質があるとすれば、むしろ一つ一つの神経細胞の正確なモデルなどほとんど必要ないかもしれません。

*1

実際、機械学習においては、ニューラルネットワークなどの生物学的なモデルとしては単純すぎると言えるようなものが非常に効果を発揮しています。

この章では、ニューロンの数学的モデルと、学習理論における数学的モデルをいくつか見ていきます。

ニューロンのモデルは様々なものが提唱されています。イオンチャネルの様子まで細かく記述したものや、ニューロンの活動電位の様子だけ記述した簡潔なものまで、いろいろです。それぞれにメリットとデメリットがあります。複雑すぎるものは、確かにニューロンの特性を正確に捉えることができますが、計算的に非常に高価です。一方で簡単な式で記述したモデルは計算的には安価で扱いやすいですが、正確さがどこまで再現できているのかと言われると、ツッコミどころも多くあります。そのような正確と計算コストのトレードオフを表したのが以下の図です。(Izhikevich E.M. (2004) Which Model to Use for Cortical Spiking Neurons? から引用。)

図 4.1 model

横軸が計算コストの高さ、縦軸がニューロンのモデルとしての正確さを表して

います。ご覧のように、様々なモデルが提唱されているのが分かります。この中で一番複雑なモデルは、一番右端の Hodgkin-Huxley モデル (HH モデル) と呼ばれるものです。HH モデルは、実験によって作られた歴史的に初めてのかなり精巧なニューロンの数理モデルで、ヤリイカの軸索を調べることによって作成されました。活動電位の単純な特性だけでなく、イオンチャネルの性質なども考慮しているので、ニューロンモデルとしての正確性は高い一方で計算コストもかなり高いです。逆に、一番簡潔なモデルは Integrated and Fire モデル (積分発火モデル:IF モデル) と呼ばれるもので、ニューロンの描写をかなり簡潔なものにすることで計算コストを抑えています。HH モデルと IF モデルの間にも様々なニューロンの数理モデルがあります。

4.2 ニューロンの数学的モデル

4.2.1 積分発火モデル (Integrated and Fire Model)

積分発火モデルは、ニューロンの活動電位の様子を簡潔に表現した代表的なモデルで、しばしば IF モデルと呼ばれます。IF モデルは、以下の膜電位 V に関する微分方程式で表現されます。

$$\tau \frac{dV(t)}{dt} = -V(t) + RI(t) \quad (\tau = RC)$$

if $V(t) \geq V_{\text{閾値}}$, *then* $V(t) = V_{\text{静止膜電位}}$

この IF モデルは、ニューロンの細胞膜を電気容量 C のコンデンサーと抵抗 R の並列回路と見て V の変化を考え、加えて並列回路の電位 V がある閾値を超えた時に V を静止膜電位にリセットするという分岐を考慮したものです。 τ は膜の時定数と呼ばれる定数で、 V の閾値と静止膜電位は任意のパラメータです。ニューロンの静止膜電位はおよそ -65mV なので、その付近に設定するのが一般的です。

この IF モデルの動作を見てみましょう。

```
using DifferentialEquations
using Plots

tau = 1.0
I = 1.0
R = 1.0

p = [tau, I, R]

function neuron(du, u, p, t)
    if u[1] < -30
        du[1] = (-u[1] + p[2]*p[3]) / p[1]
```

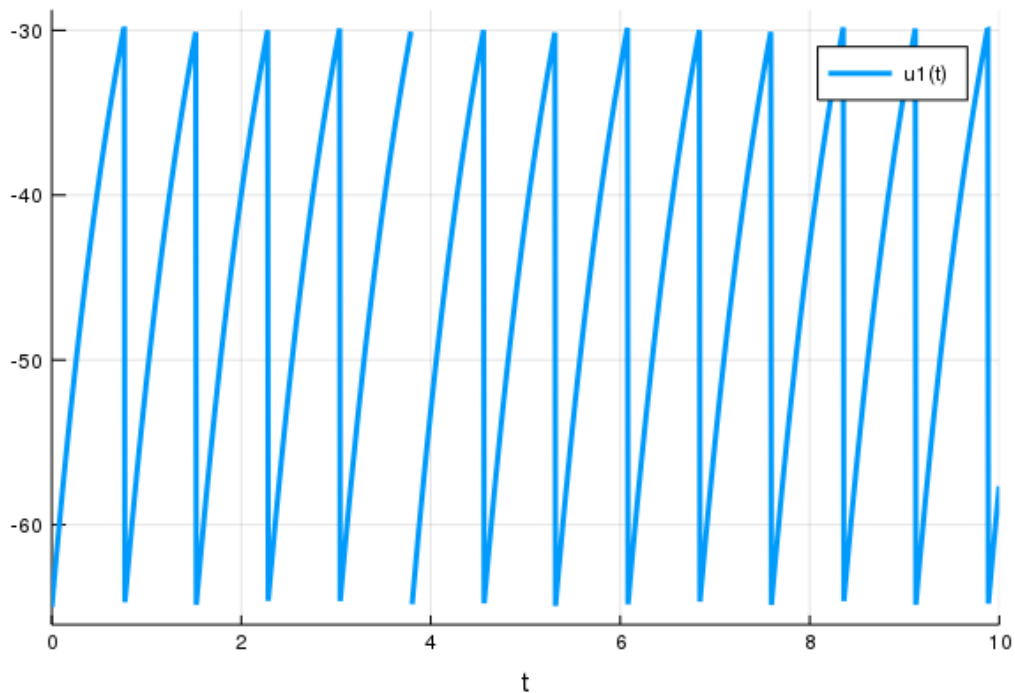


図 4.2 IF

```

else
    u[1] = -65
end
end

u0 = [-65.0]
tspan = (0.0,10.0)
prob = ODEProblem(neuron,u0,tspan,p)
sol = solve(prob,reltol = 1e-8)

plot(sol)
savefig("I&F.png")

```

4.3 ニューロンの数学的モデル

4.4 脳の学習理論のモデル

脳の学習は、シナプス艦の結びつきの強さの変化によって行われていると考えられています。ここでは、シナプスの結びつきの振る舞いを数学的に記述していきます。

第 5 章

Neuromorphic Computing

5.1 概要

Neuromorphic Computing というのは，脳の計算処理能力を模倣した人工的なチップ（物質）を作り上げようとする研究領域です。従来のコンピュータと異なる仕組みを持つ，新たな物理的な計算処理デバイスを作り上げようとするものです。いわゆる機械学習と呼ばれる技術とは決定的に異なる点は，機械学習が従来のノイマン型コンピュータ上における脳の計算機能のシミュレーションであるのに対して，Neuromorphic Computing は物理的に脳の計算処理機能をもったハードウェアを作ろうとする，脳のエミュレーションであるということです。この点が決定的に異なります。

脳の計算機能において特徴的なものは，アナログ計算，エラー耐性，event-driven，並列計算，リアルタイム処理，超低消費電力，ハードウェアとソフトウェアの一体化，といったものがあげられます。この中で，社会的に強い要請があるのは超低消費電力です。従来のノイマン型コンピュータは，その仕組み上何度もデータを CPU とメモリの間で行き来する必要があります。このデータの転送にかかる電力消費が，実際の計算にかかる電力消費よりもかなり大きいのです。^{*1}加えて，コンピュータを起動している間は重たい計算をしていなくても常に待機電力として大きなエネルギーが消費されています^{*2}。また，データを行き来させる間の CPU の待機時間が計算処理速度のボトルネックにもなっており，これも大きな問題として知られています。（フォンノイマンボトルネック）一方，脳的な計算ではシナプスという構造自体が計算処理とメモリ機能を同時に果たしており，CPU とメモリ間のボトルネックのようなものは存在しません。脳全体にメモリが分散されてい

*1

*2

るのです。また、原理的に発生する消費電力は、外部から入力信号を受けた時のみで、待機中は何も動作しないので消費電力は0です。計算処理の際の消費電力も従来のコンピュータに比べ圧倒的に少なく済みます。

計算処理のリアルタイム性も重要なポイントです。普段の生活から分かるように、脳をもつ我々生物は外部の自然環境にリアルタイムに適応しています。歩いたり走ったりするたびにいちいち経路の計算をするようなことはありません。これは、神経細胞が物理的に繋がりを持っていること自体が計算機能としての役割を果たしているため、ほとんどノータイムで外部の状況を計算処理することができるためです。一方で、従来のコンピュータはデータを入力してから電子回路上で電子を操って外部状況を逐一計算する必要があります。そのため、原理的に外部の状況をつねにリアルタイムで計算し続けるというのは非常に厳しく、達成するためには高性能な計算機、冷却器などが必要になってきます。高負荷な3Dゲームのコマ落ちなどは分かりやすい例かもしれません。現実世界は非常に高解像度な3Dの世界ですが、処理落ちして世界が一時的に止まるなんてことはありません。しかし、コンピュータ上に再現された3Dのゲーム空間ではそれがありません。なぜなら、3Dゲームは、現実世界をコンピュータ上で計算によってシミュレート・再現したものであり、計算するのに時間がかかるからです。

一方で、注意しておかなければならない点もいくつかあります。もっとも重要な指摘の一つとしては、生物の脳は汎用的な計算機ではなく、ある特定の目的に特化した計算機であるということです。(not general, but dedicated) 脳は、機能ごとによりかなり明確に領域が分かれており、その領域ごとに非常に明確な達成目標があります。例えば、眼の網膜から後頭部の第一視覚野と言われる領域は、視覚の明暗やエッジ検出という目的を達成するために非常に複雑な回路を形成しています。実際、脳の機能を考える上では、身体の機能も同時に考えることが不可欠です。なぜなら、脳は入力に対して、なにかしらの形で出力をするための機構だからです。この理由から、Neuromorphic Computingは従来のコンピュータをすべての点で上回るのかという疑問に対して、明確にNOと答えることができます。情報の正確な処理、半永久的な記憶、あらゆるデバイス間での通信、といった点でNeuromorphicは従来のコンピュータに敵うことはありません。この原因は、情報の取り扱い方がデジタルなのかアナログなのかということに由来します。デジタルな情報処理は基本的に全く同じ情報を再現することが可能ですが、アナログな情報処理では不可能です。これは、磁器のビデオテープのコピーとDVDのコピーを

考えればわかります。磁器はアナログデータですから、どれだけ正確にコピーしようとしても、時間が経つにつれて徐々に情報が変化してしまいます。一方で DVD データはデジタルな 0 と 1 による表現なので、原理的にはいつまでもデータを保存することが可能なのです。アナログであるというのは、連続値であるということ、現実世界そのものであるということと言えます。デジタルであるというのは、離散値であるということ (0 と 1)、現実世界から離れた時間で動作するもの (CPU のクロック) と言えます。この違いも非常に重要です。

5.2 アナログかデジタルか

5.3 学習能力をどのように獲得するか

5.4