

ディープラーニング入門 I

ニューラル・ネットワークの基礎



生物のニューラル・ネットワークと脳

ディープラーニング入門 I

ニューラル・ネットワークの基礎





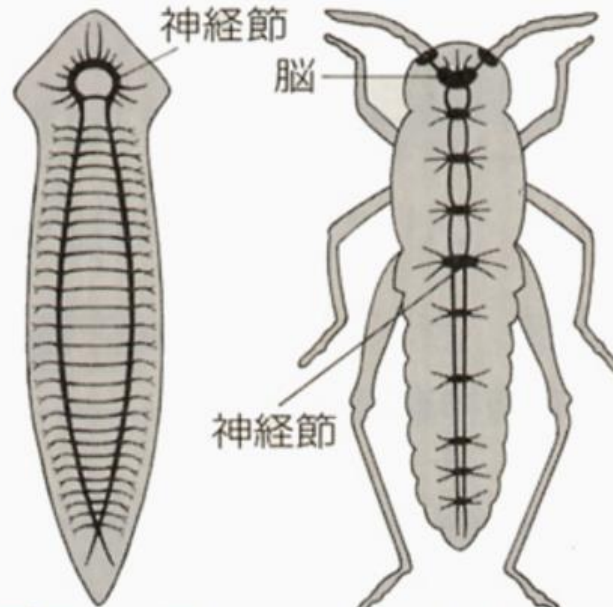
生物のニューラル・ネットワークと脳



クラゲだって神経はある

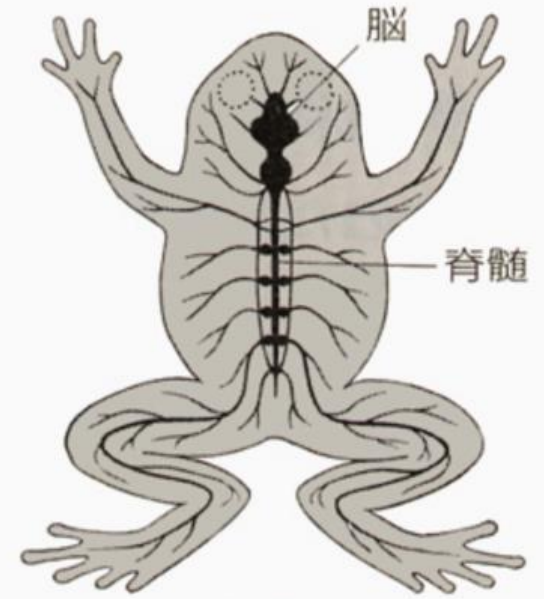


ヒドラ
(散在神経系)



プラナリア
(かご形神経系)

バッタ
(はしご形神経系)



カエル
(管状神経系)



神経系の起源と進化

脳の出現

<https://cns.neuroinf.jp/jscpb/wiki/神経系の起源と進化>

もう少し詳細に神経系の進化を見てゆくと、神経細胞が最初に出現したと思われる腔腸動物の散在神経系では、神経細胞の特殊化はまだあまり進んでいないが、すでに中枢神経系の萌芽もふくめ神経系の基本的な成分は全て揃っている。集中神経系の始まりとして、左右相称動物の根元に位置する扁形動物のプラナリヤでは、脳形成の遺伝子プログラムはすでに揃っていて、脳の基本となる構造を作るためのロジックは進化のかなり初期段階ででき上がっていたのでないかと思われる。

最近の分子系統学の進歩により、前口動物は、従来のものとはまったく異なる二大グループに分けられて考えられるようになってきている。それが、脱皮動物と冠輪動物である。冠輪動物の方は、軟体動物・環形動物・腕足動物・扁形動物などを含み、脱皮動物の方は節足動物・線形動物を含む。従来の節構造の進化についての(神経系についても同様であるかも知れない)環形動物から節足動物の進化の考えは、変更が必要かもしれない。前口動物の一大グループである冠輪動物の多様な神経系を比較すると、イカやタコなどの頭足類の素晴らしい脊椎動物様の巨大脳に行く。



神経系の起源と進化

脳の出現

<https://cns.neuroinf.jp/jscpb/wiki/神経系の起源と進化>

前口動物の脳進化のもう一つの頂点は、地球上で最も繁栄している動物群である昆虫の微小脳である。これは、「小型・軽量・低コストの情報処理装置の傑作」である。

一方、後口動物の神経系の進化は、まず、ウニ、ナマコ、ヒトデなどの無脊椎動物、棘皮動物から始まる。この神経系では、原始的な中枢制御機能をもつ介在神経系が出現する。それでもしっかりした脳構造がはっきりしないので、この神経系を散在神経系と呼ぶ研究者も多い。中枢制御装置として神経環を持つので、腔腸動物のクラゲと一緒にして環状神経系と呼ぶ研究者もいる。

系統樹では、この棘皮動物門は、半索動物門(ギボシムシ)を経て脊索動物門の尾索類(ホヤ)・頭索類(ナメクジウオ)・脊椎動物と続く。棘皮動物と半索動物は散在神経系を持ち、脊索動物のみが前後に伸びた中枢神経系を持つと言われる。頂点の脊椎動物の背側神経系の起源になる背側神経管の出現は、尾索類(ホヤ)の幼生で観察される。最終的には、哺乳類、さらにはわれわれヒトの脳皮質の発達した脳にたどり着く。



NEURONS AND SYNAPSES
 This wiring diagram shows more than 4,500 of the 8,000 neuron-to-neuron connections, or synapses, in the worm's nervous system. Each dot represents a single neuron:

- **SENSORY NEURONS**
detect external stimuli
- **CONNECTOR NEURONS**
relay signals
- **MOTOR NEURONS**
control muscle movement

線虫のC. Eleganceは、すべての神経の接続がわかっている唯一の生物である。
302の神経と8,000のシナプスがある。

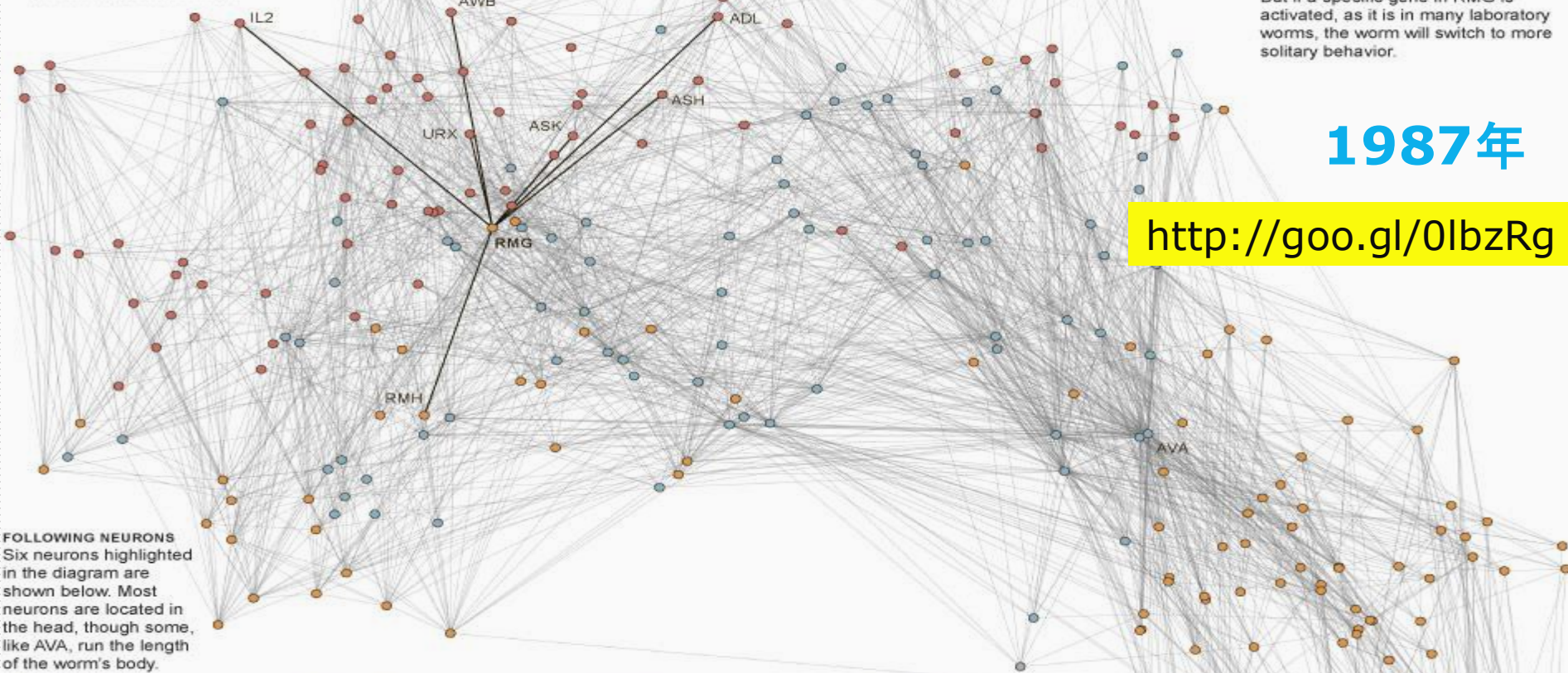
SOCIAL AND SOLITARY WORMS
 Most worms in nature congregate in clumps, a behavior controlled by two neurons called RMG, one of which is highlighted at left.

RMG combines information from several sensory neurons, also highlighted, and then signals the worm's muscles to move toward nearby worms if conditions are right.

But if a specific gene in RMG is activated, as it is in many laboratory worms, the worm will switch to more solitary behavior.

1987年

<http://goo.gl/0lbzRg>



FOLLOWING NEURONS
 Six neurons highlighted in the diagram are shown below. Most neurons are located in the head, though some, like AVA, run the length of the worm's body.



NEURON: URX
 Senses oxygen in the environment



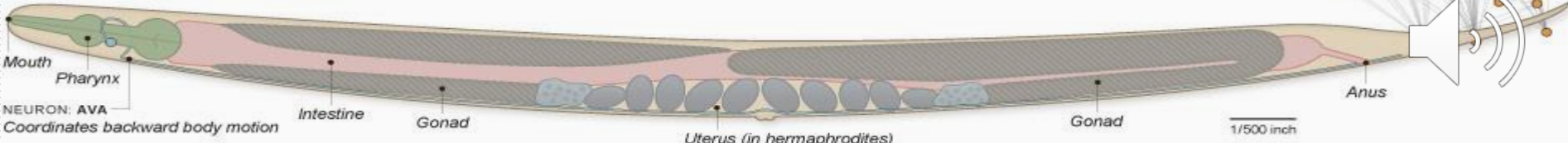
ASH and ADL
 Sense poisons and other noxious stimuli



ASK
 Senses food signals and detects sex pheromones from other worms



RMG
 Hub of a network of neurons that controls social behavior



NEURON: AVA
 Coordinates backward body motion



WormBookとWormBase

- **WormBook** is an open access, comprehensive collection of original, peer-reviewed chapters covering topics related to the biology of the nematode worm *Caenorhabditis elegans* (*C. elegans*). WormBook also includes WormMethods, an up-to-date collection of methods and protocols for *C. elegans* researcher

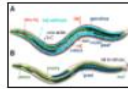
<http://www.wormbook.org/>

- **WormBase** is an online biological database about the biology and genome of the nematode model organism *Caenorhabditis elegans* and contains information about other related nematodes

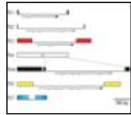
- **Sydney Brenner** は、こうした業績で、2002年ノーベル賞受賞



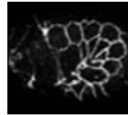
C. elegans



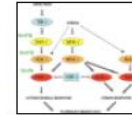
Introduction



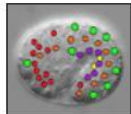
Genetics and genomics



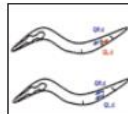
Developmental control



Signal transduction



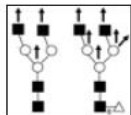
Molecular biology



Post-embryonic development



Neurobiology and behavior



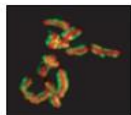
Biochemistry



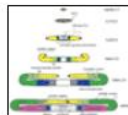
Sex determination



Evolution and ecology



Cell biology



The germ line

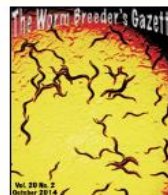


Disease models and drug discovery

Other nematodes



WormMethods




The Worm Breeder's Gazette
Last updated March 7, 2021



WormHistory





視覚の誕生と
生物のニューラル・ネットワークの飛躍 

「視覚」の誕生と発展

- 生物の進化の歴史の中で、「視覚」の能力は、5億年前の「カンブリア紀」に獲得された。それは、長い長い歴史を持っている。当時の食物連鎖の頂点に立っていた「アノマロカリス」は、立派な眼を持っていた。
- 「眼」の獲得、同じことだが、「視覚」の能力の獲得は、「捕食者」にとっても（獲物を見つける）、「被捕食者」にとっても（捕食者から逃れる）、その「生き残り」に、死活的に重要な意味を持つことになった。人間を含めて、すべての動物で、強い淘汰の圧力のもとで、視覚処理の能力は、高度に発達している。
- 同時に、この「視覚」の能力は、追いかける、捉える、あるいは、逃げるといった「運動能力」に深く結びついている
- それは、生物のニューラル・ネットワークの進化の飛躍をもたらした。



アノマロカリス



多くの生物は目を持っている



ゲーリングによる「目の遺伝子」Pax-6の発見

https://www.brh.co.jp/seimeishi/journal/012/ss_1.html





視覚と運動能力は、強く結びついている

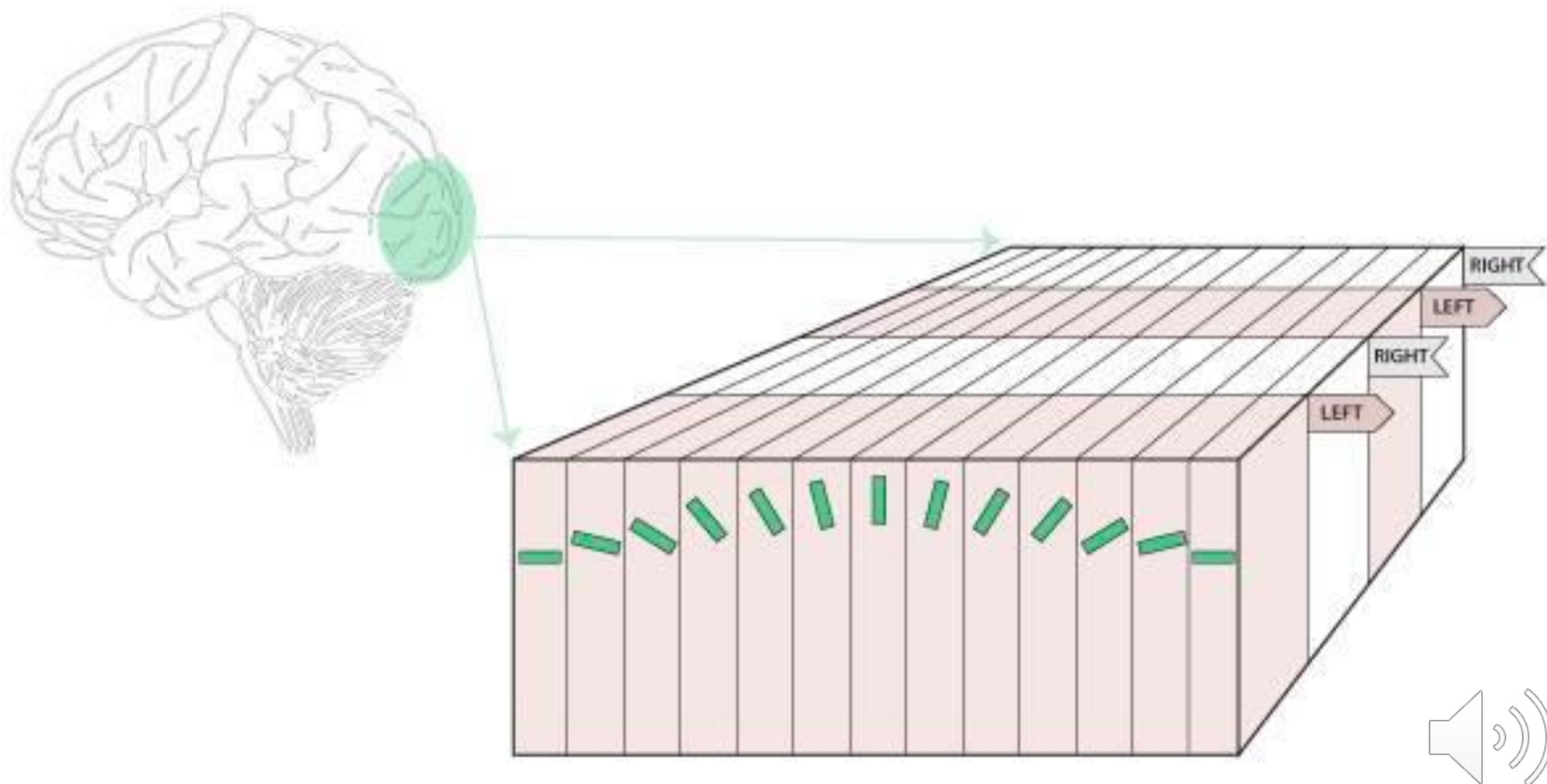




センサーを持つマシンの自律的運動
ディープラーニングが得意な分野



60年代から70年代にかけての、Hubel と Wieselの
大脳視覚野の研究は、各方面に大きな影響を与えた。
ニューラル・ネットワークの最初の研究も、こうした影響の
もとで始まった。





1960年代

David H. Hubel

Torsten Wiesel



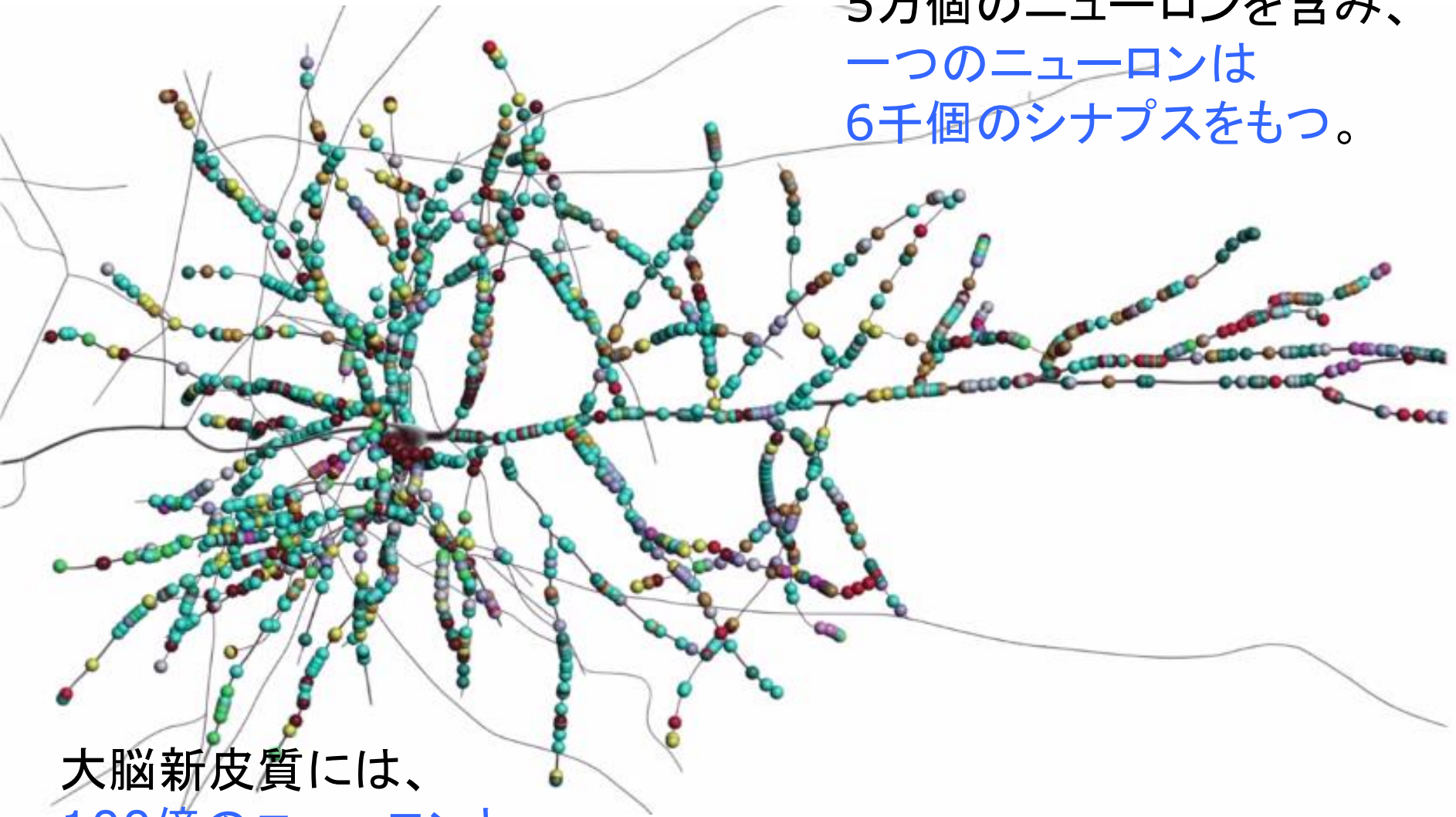
Nobel Prize in Physiology or Medicine (1981)



2012年「Googleの猫」



脳は、1mm立方に
5万個のニューロンを含み、
一つのニューロンは
6千個のシナプスをもつ。

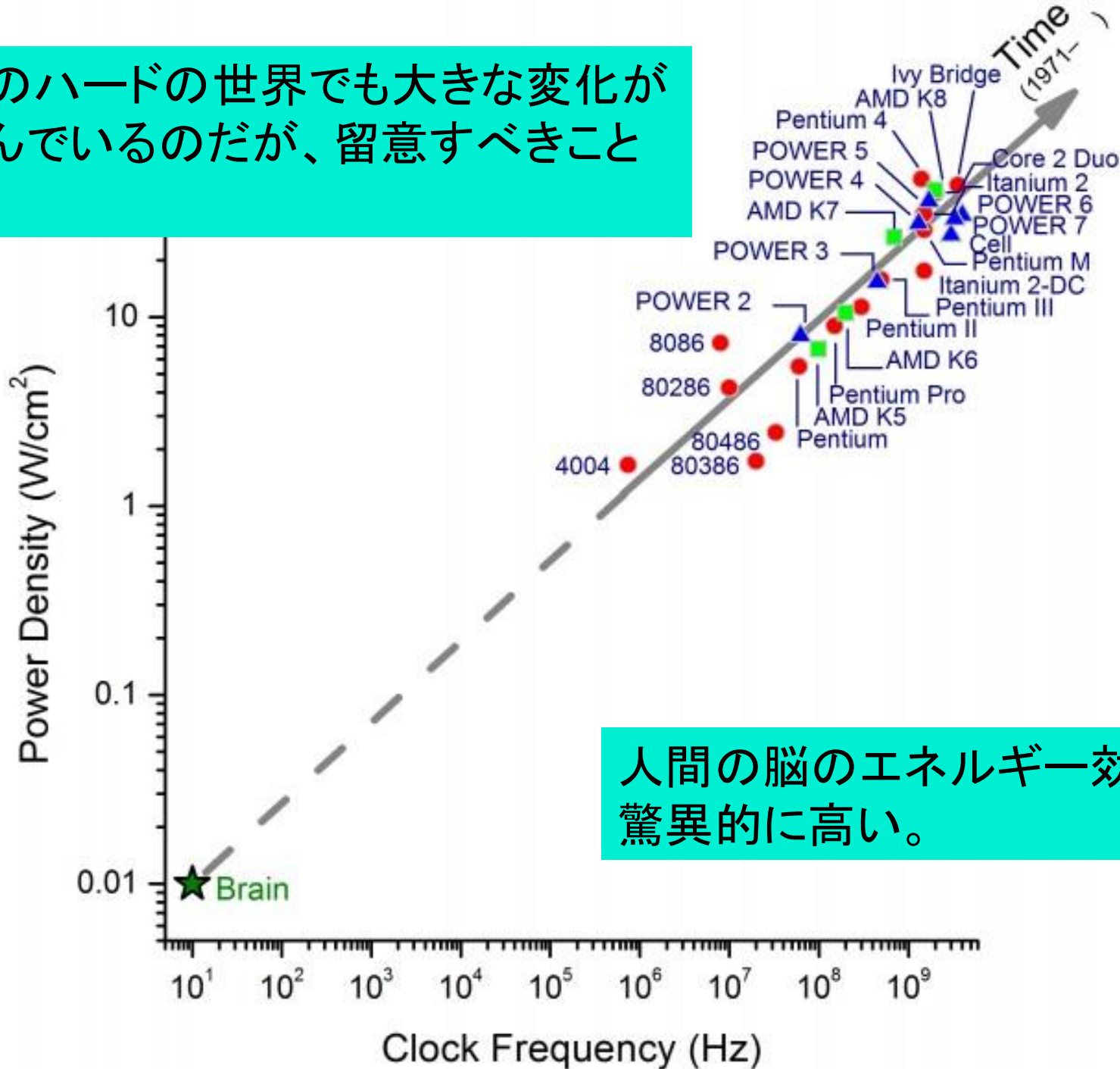


大脳新皮質には、
100億のニューロンと
60兆個のシナプスがある！



AIのハードの世界でも大きな変化が進んでいるのだが、留意すべきこと

.....



人間の脳のエネルギー効率は、驚異的に高い。





「人間と機械」という対比だけでなく、「生物と機械」という対比でも、考えることは面白い。

ニューラル・ネットワークが、今後、大活躍するであろう「感覚-運動系」でも、生物の能力は、驚異的である。





脳研究の取り組み



脳研究の取り組み

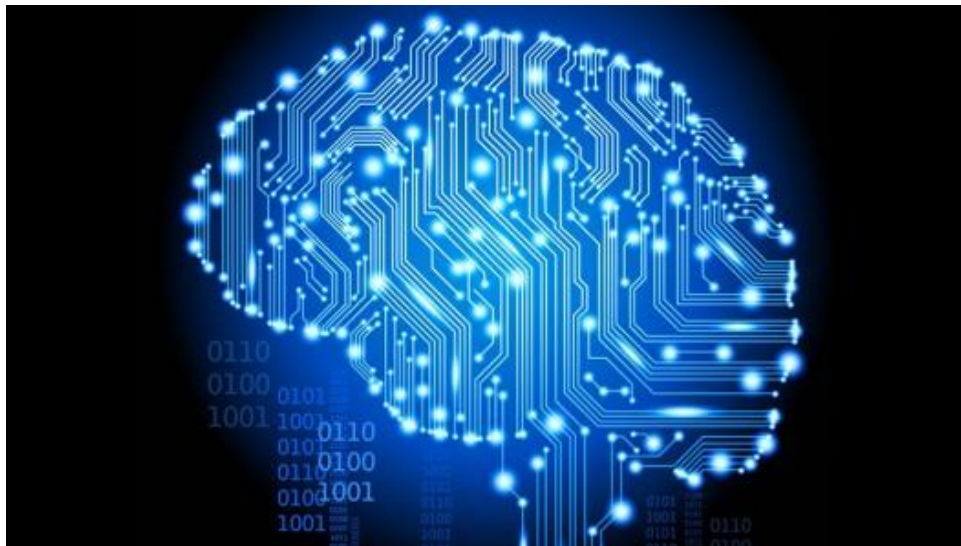
現在は、それぞれ異なった道を歩いているのだが、ニューラル・ネットワークの研究と脳研究とは、「人間の知能」の解明という共通の目標で結ばれている。ここでは、脳研究のいくつかの取り組みを紹介する。



Blue Brain Project

Henry Markram

- ❑ “I wanted to **model the brain** because we didn’t understand it.”
- ❑ “The best way to figure out how something works is to **try to build it from scratch.**”

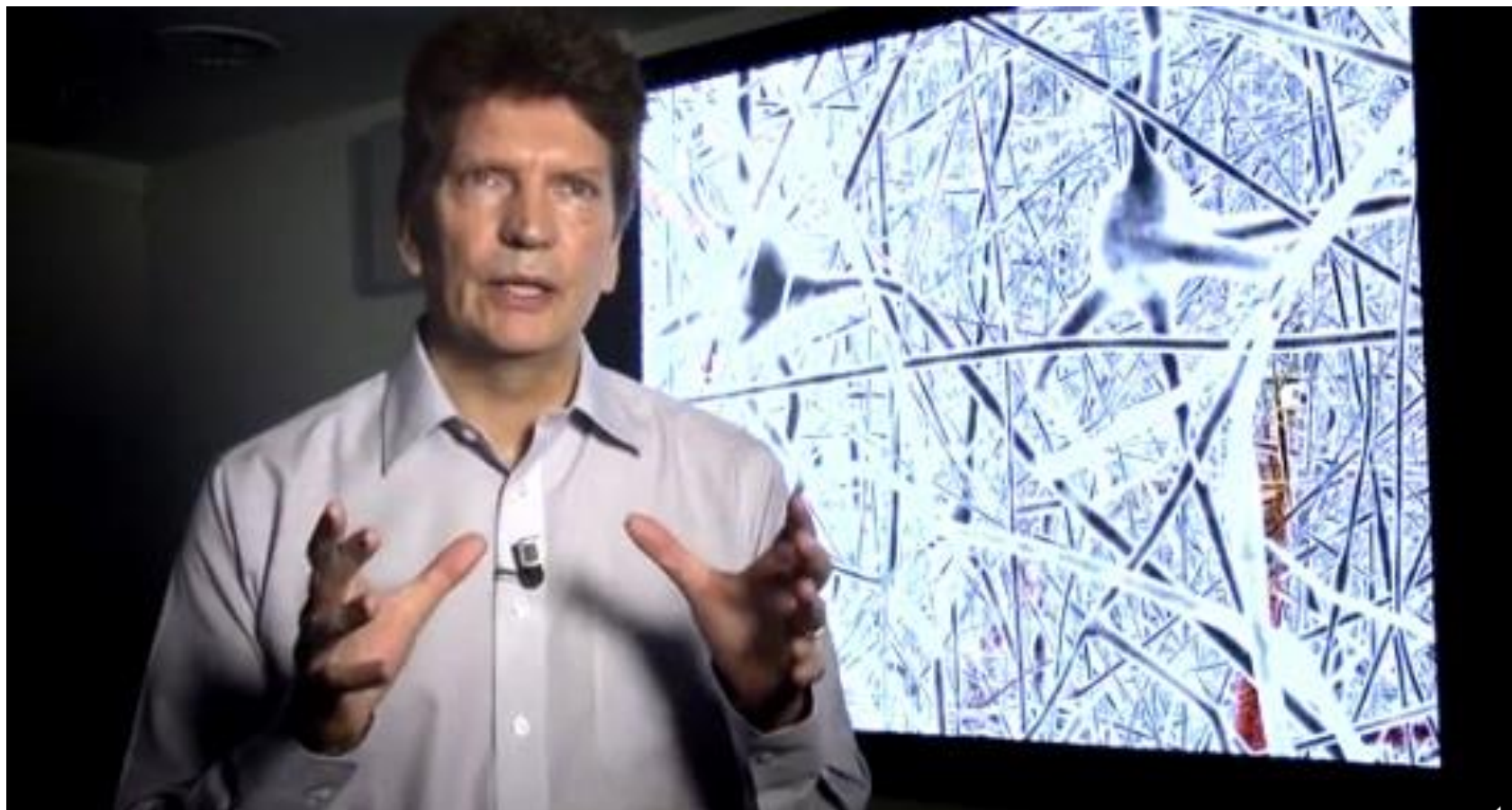


in vivo

in vitro

in silico 

EU Human Brain Project



2013年10月 10年間で総額約12億ユーロ



EU Human Brain Project

ビジョン

人間の脳を理解することは、21世紀の科学が直面している最も偉大な挑戦の一つである。もしも、我々が、それに対して立ちあがる事が出来るならば、我々は、我々を人間にしているものが何であるかについて深い洞察を得て、革命的なコンピュータ技術を構築し、脳の異常に対して新しい治療法を開発出来るだろう。今日、初めて、現代のICT技術が、こうした目標を到達可能なものになっている。



EU Human Brain Project

批判 ECへの公開質問状

Open message to the European Commission concerning the Human Brain Project

Sign the letter

Summary. Neuroscience advances our understanding of normal and pathological brain function, offering potentially enormous benefits to society. It is, therefore, critical to Europe. The Human Brain Project (HBP), sponsored by the European Commission (EC), was meant to forward this mission. However, due in great part to its narrow focus, it has been highly controversial and divisive within the European neuroscience community and even within the consortium, resulting in on-going losses of members. The HBP is now scheduled for review and we wish to draw the attention of the EC to these problems. We believe the HBP is not a well conceived or implemented project and that it is ill suited to be the centerpiece of European neuroscience. We are particularly concerned about the plan to tie a substantial portion European member states' neuroscience funding to the HBP through so-called 'partnering projects'. We call for the EC to go beyond the strict requirements of the upcoming review, to demand transparency and accountability and, if necessary, change the structure of the HBP's governance and supervision to correct their shortcomings. Failing that we call for the EC to redirect the HBP funding to smaller investigator-driven neuroscience grants. We stand fully behind a strong and united European neuroscience strategy and we pledge not to seek funding through HBP partnering projects that would compromise that mission.

[Official Response by the EC - 18 July 2014](#) | [Press Releases](#)

Read the full letter



脳研究をめぐる「対立」の構図

認知科学 vs 脳科学、トップダウン vs ボトムアップ

- 発端は、次年度のプロジェクトの予算配分で、認知科学的なアプローチの予算が、ばつさりと切られたことにあるらしい。それに反発した研究者のグループが、150名の連名で、欧州委員会に公開質問状を提出した。
- 対立の根底にあるのは、脳研究でのアプローチの違い。Henry Markramら主流派は、ニューロンとシナプスの数学的モデルに基づいて脳全体のモデルを**ボトムアップ**に作り上げようというアプローチ。一方、反対派は、脳研究には、認知科学の知見に基づいた**トップダウン**のリバーズ・エンジニア的なアプローチが必要だという。
- 「鳥の羽の全てをシミュレーションしたとしても、鳥が空を飛べることを解明出来ないのと同じことだ」



NIH Human Connectome Project

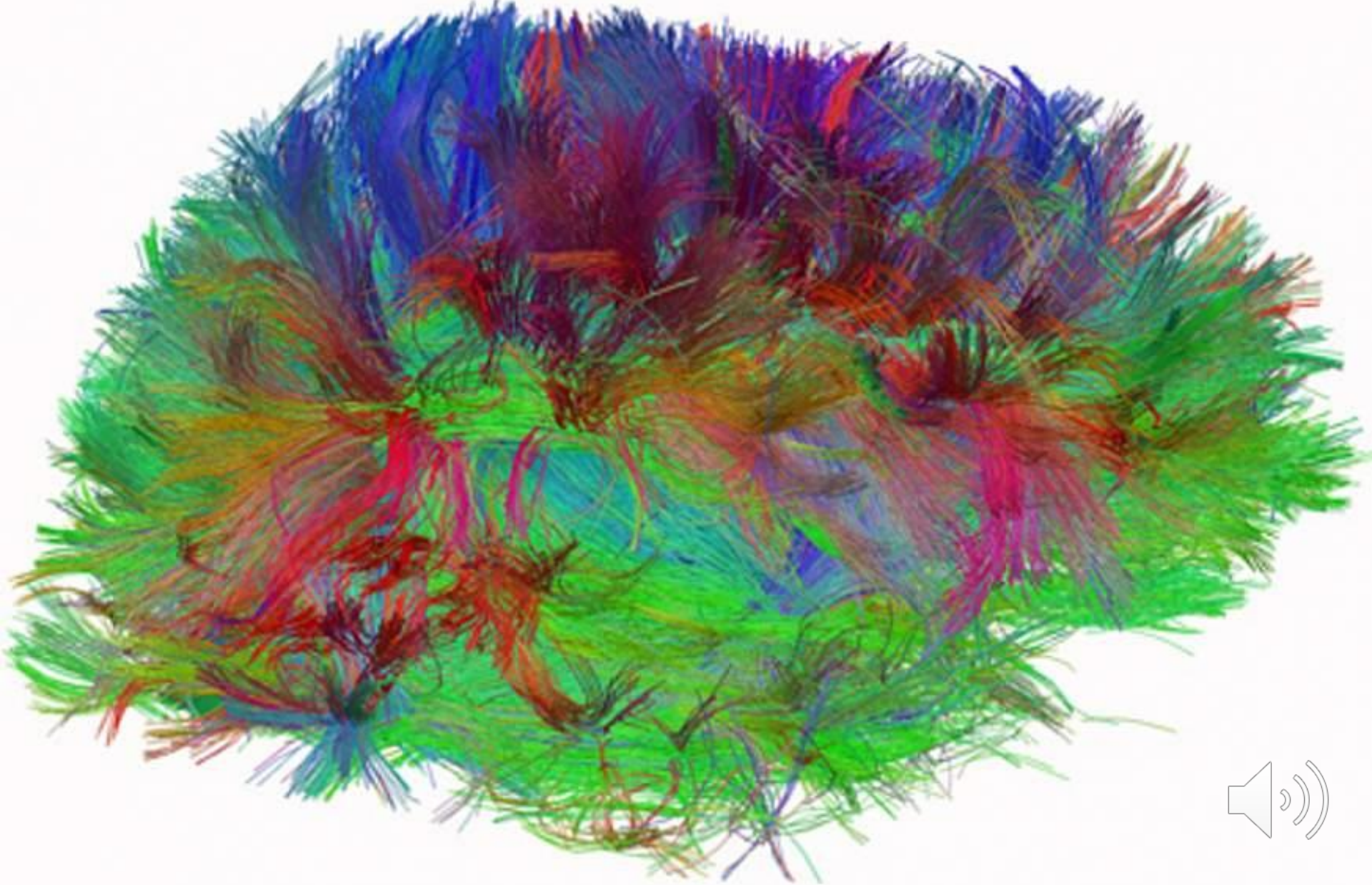


2010年10月 40億円の賞金

<http://www.nih.gov/news/health/sep2010/nimh-15.htm>



\$40 million awarded to trace human
brain's connections



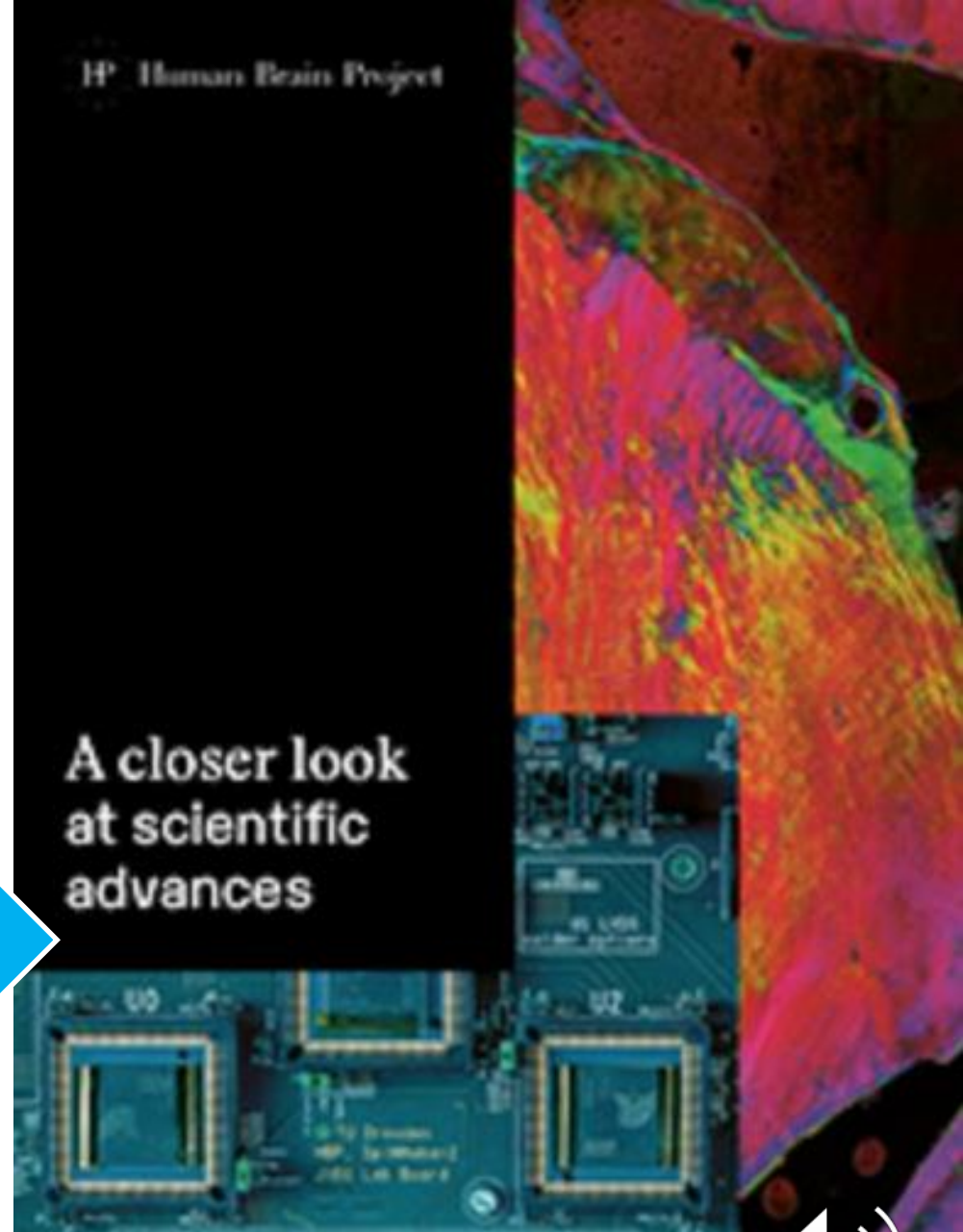


脳研究の現在



Human Brain Project (EU)

Learn about the research
being done by the
Human Brain Project in
the brochures.



<https://www.humanbrainproject.eu/en/science-development/scientific-achievements/brochures/>



p.06

脳の計算パワー
への小さな秘密

p.07

人間の脳の
最も詳細な地図

p.05

脳を覗き見る
新しい方法

p.09

接続性を
測定する

p.15

AIのエネルギー効率
を改善するために、
脳を模倣する

p.10

デジタル脳モデル
を用いたてんかん
手術の改善

p.13

脳にインスパイア
された動作と認識
を持つロボットを作る

p.12

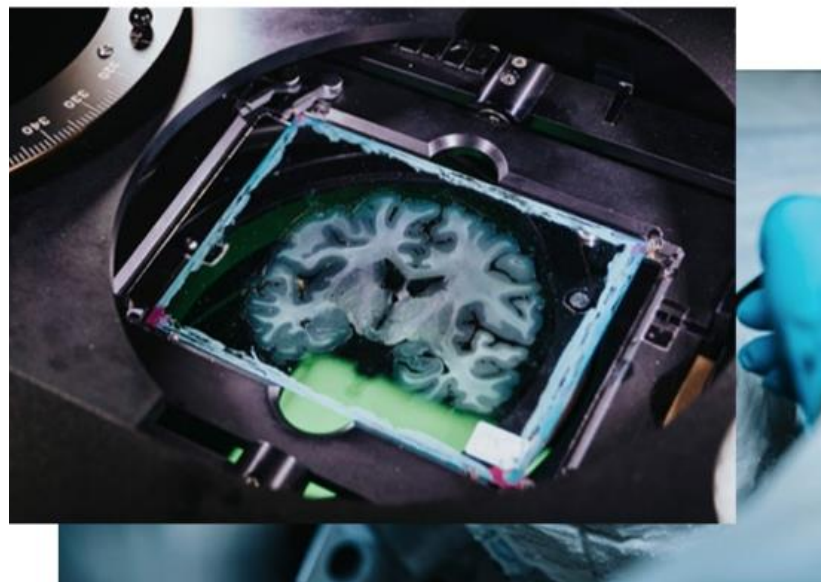
盲目や麻痺を
克服するニューロン
インプラント



New methods to look into the brain

Watch Video

イタリア、ドイツ、英国のチームを含むHBPの研究者たちは、脳を極めて詳細に画像化する新しい高解像度手法を開発した。これには、無傷のマウスの脳を初めて細胞以下の解像度でイメージングできるようにした技術や、異なる生物種のコネクトームを極めて高解像度で比較できる新しい手法(写真)などが含まれる。並行して、膨大な量の脳データをスーパーコンピュータ上で迅速に解析するための、新しいディープラーニング手法も確立された。



1an brain section in a
rized Light Microscope



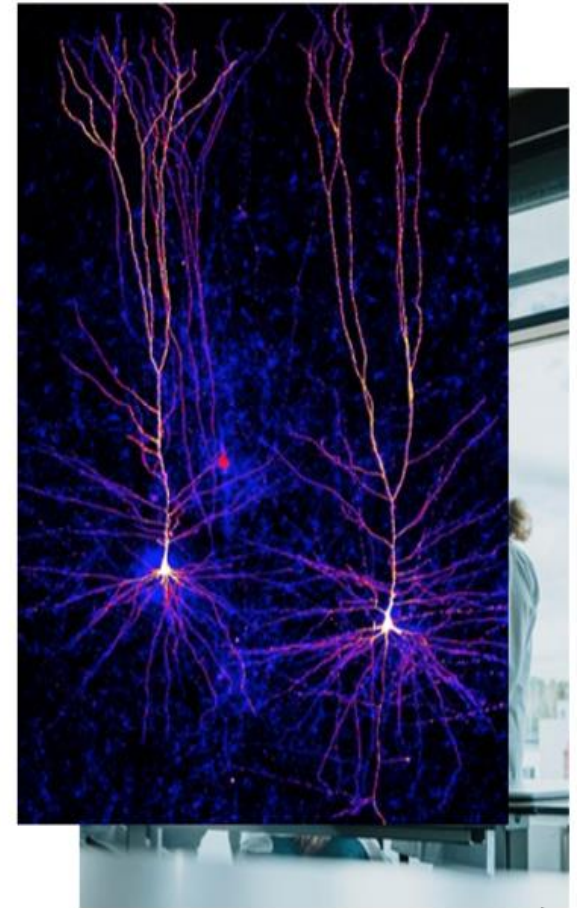
The tiny secret to the brain's computing power

神経細胞には樹状突起と呼ばれる長い木のような突起があり、電気化学信号を送受信する。ドイツのHBPの研究者たちは、ヒトの場合、ある種のニューロンの樹状突起が、脳のユニークな処理能力の秘密を握っている可能性があることを発見した。

その予想外に複雑な活動によって、以前は多層の神経回路網を必要とすると考えられていた計算問題を、単一の神経細胞が解くことができるのだ。

Layer 2/3 neurons of the human neocortex

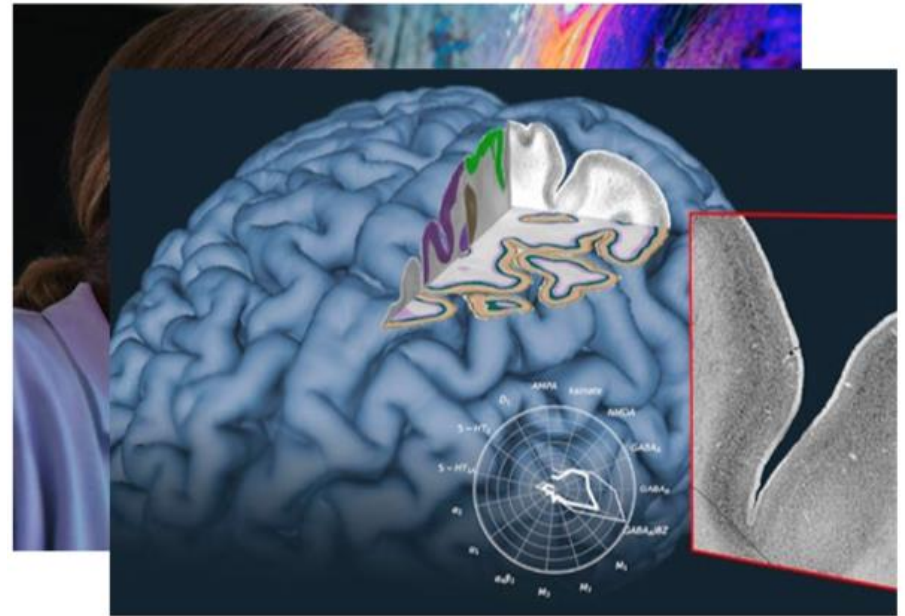
[Gidon et al. Science 367, 83–87 \(2020\). → Press release](#)



The most detailed atlas of the human brain

EBRAINS Atlas

ドイツのHBP研究者が、これまで確認されたことのないほど多くの脳領域のマップを含むヒトの脳のアトラスを作成した。このアトラスは、脳の細胞構造を3次元空間で示し、個々の脳間のばらつきを反映している。膨大なデータはスーパーコンピュータとAIを必要とする。アトラスはEBRAINS上で公開されており、常に新しいデータが統合されている。これにより、研究者は共同で脳を解読することができ、臨床医は神経疾患に苦しむ患者の治療法を改善することができる。

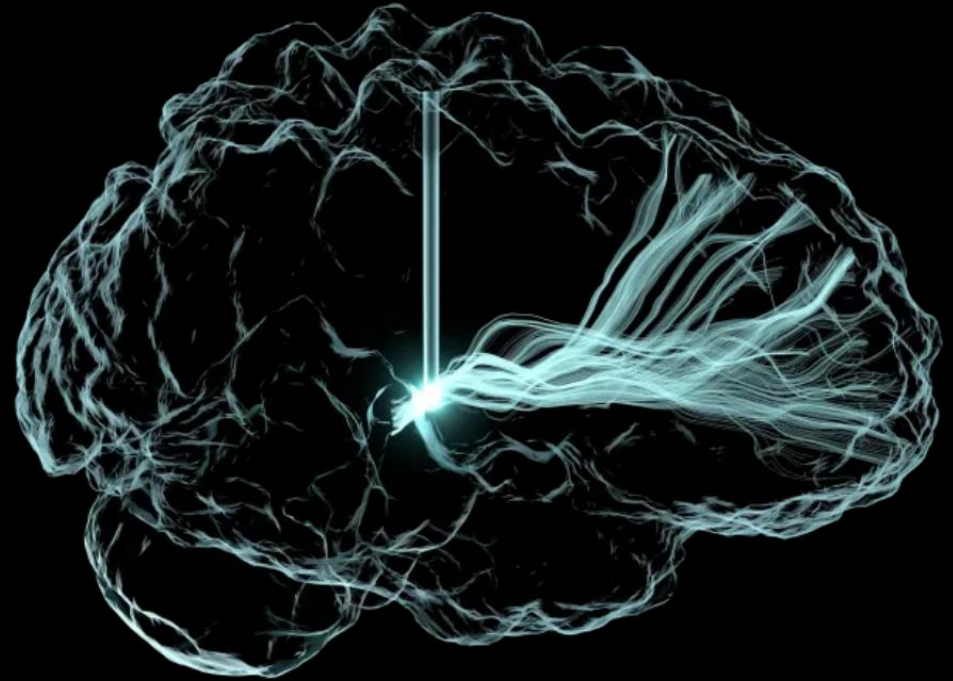


The multi-level human brain atlas of the HBP

Brain Initiative (US)

The *Brain Research Through Advancing Innovative Neurotechnologies*[®] (BRAIN) Initiative

Revolutionizing our understanding
of the human brain



<https://braininitiative.nih.gov/>

LEARN MORE

Image Courtesy of Andrew Janson, University of Utah Scientific Computing and Imaging Institute (Show Us Your BRAINs! Photo & Video Contest 2019)



Simian Symphony

音楽にリアルタイムに反応する海馬のニューロンの映像



https://www.youtube.com/channel/UC_muVZ3XF1aCNtM_YfcCyyg/videos



脳の理論モデル

Fristonらの「脳の理論モデル」の研究は、Markramのようにコンピュータ上に脳のモデルを作ろうというものでも、それに反対した認知科学者たちのアプローチとも異なるものだ。

それは、「脳のモデル」を構築する上での原理的な方法論を明確にしようという問題意識を、彼らが持っていることだ。Fristonはそれを「最小自由エネルギー原理」と呼ぶ。

全てをそれで説明しようとするのは難しさもあるかもしれないのだが、その立場は、Prigogine や Jaynesといった、20世紀の偉大な達成を忠実に受け継ごうとしているように、僕には見える。



ニューロンの働き

ニューロンはいつ発火するのか？

ディープラーニング入門 I

ニューラル・ネットワークの基礎





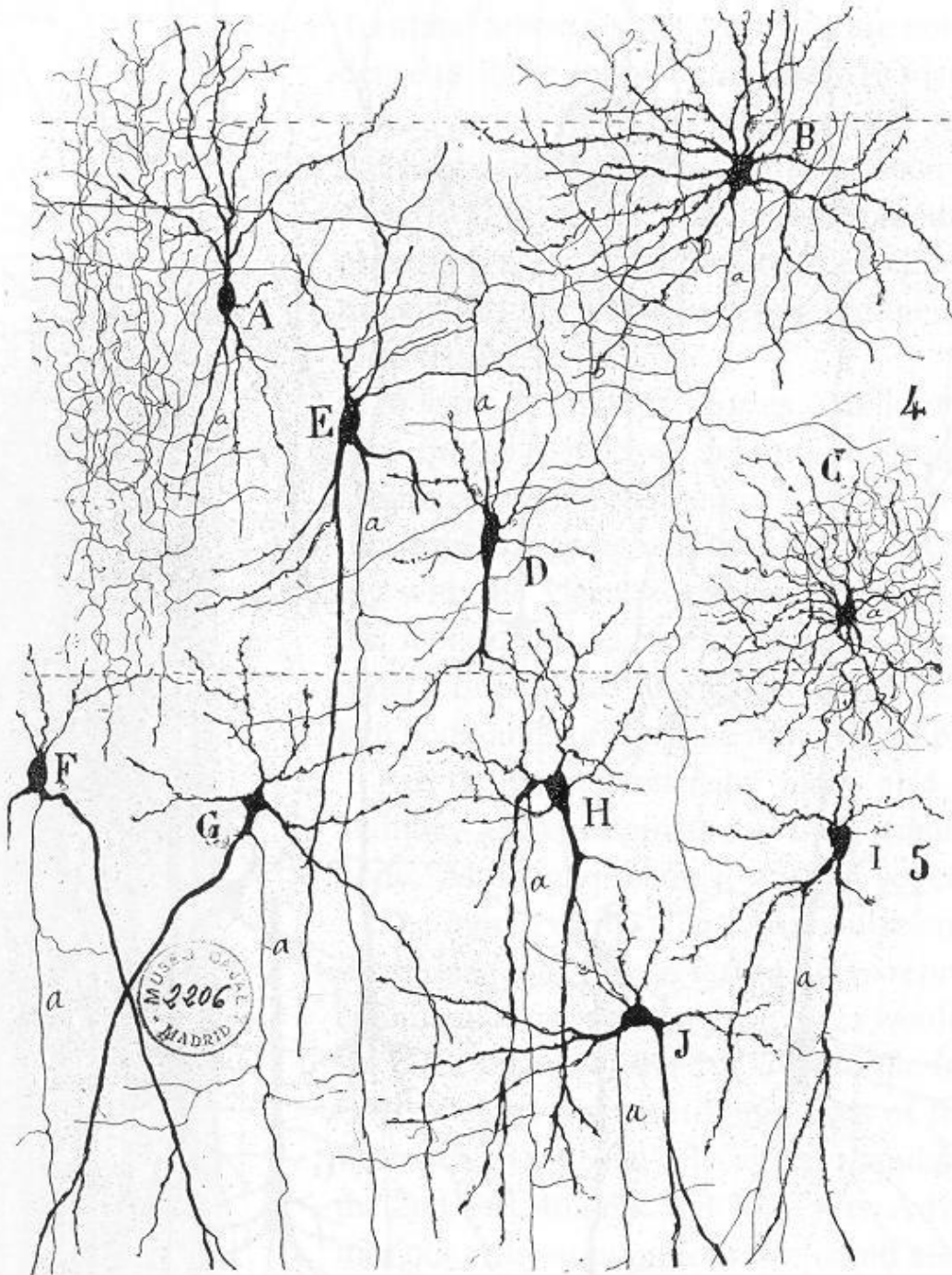
ニューロンの働き



ニューロンの働き

脳の働きも、最終的には、脳を構成する無数のニューロンの働きに帰着する。



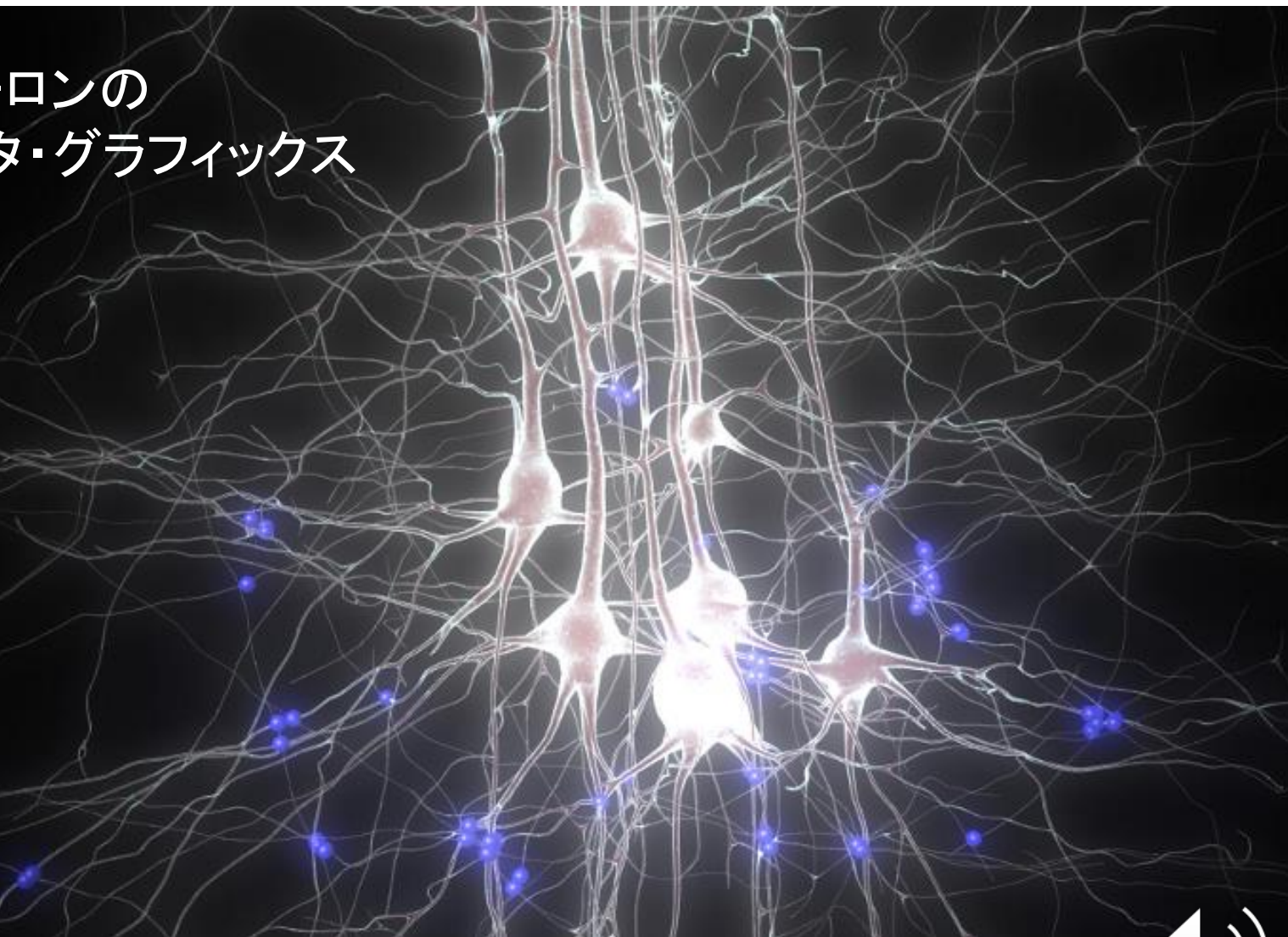


図は聴覚野における神経細胞の形態の多様性を示している。

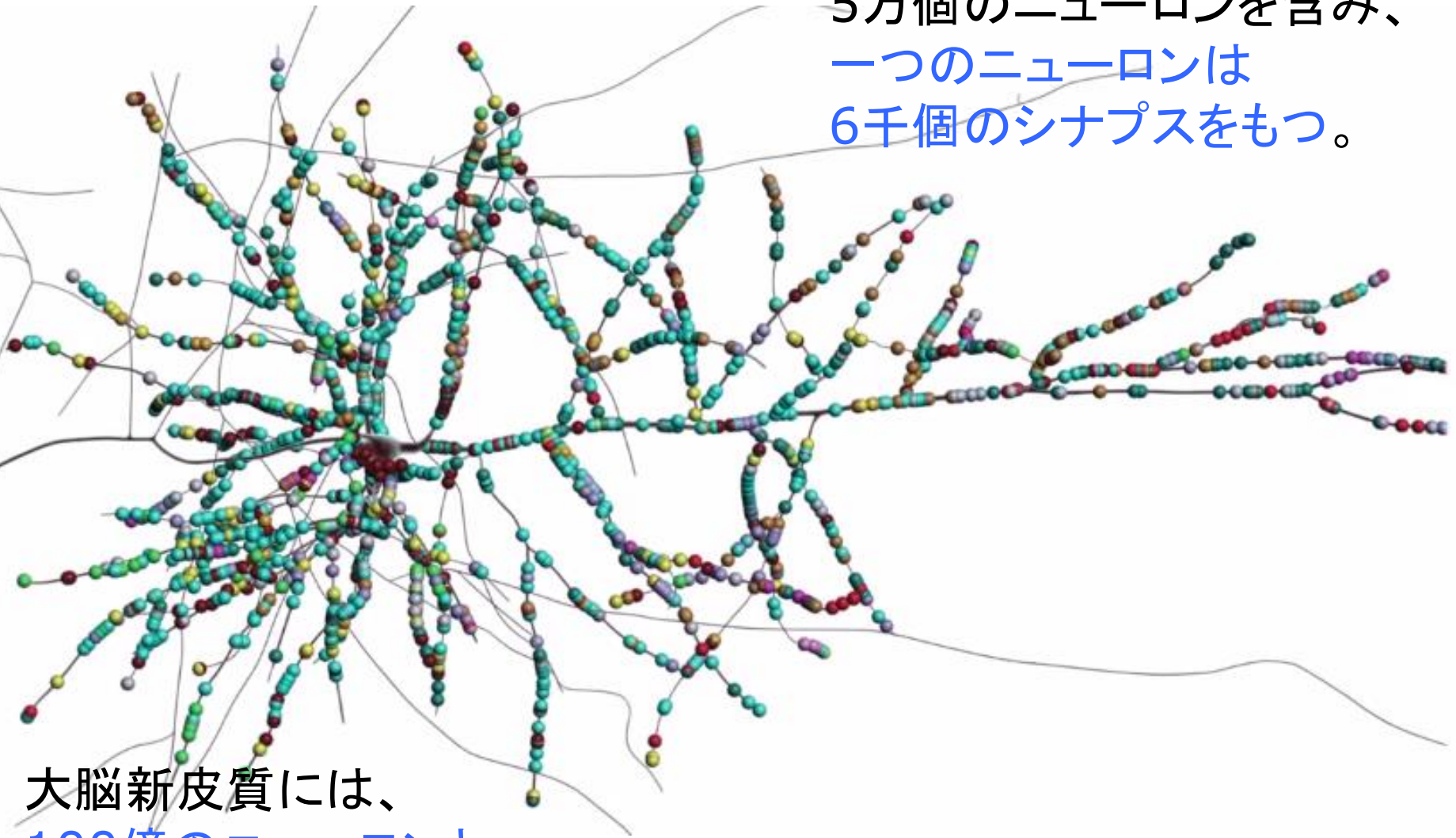
https://en.wikipedia.org/wiki/Neural_circuit



脳のニューロンの コンピュータ・グラフィックス



脳は、1mm立方に
5万個のニューロンを含み、
一つのニューロンは
6千個のシナプスをもつ。



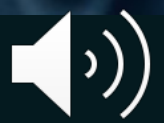
大脳新皮質には、
100億のニューロンと
60兆個のシナプスがある！



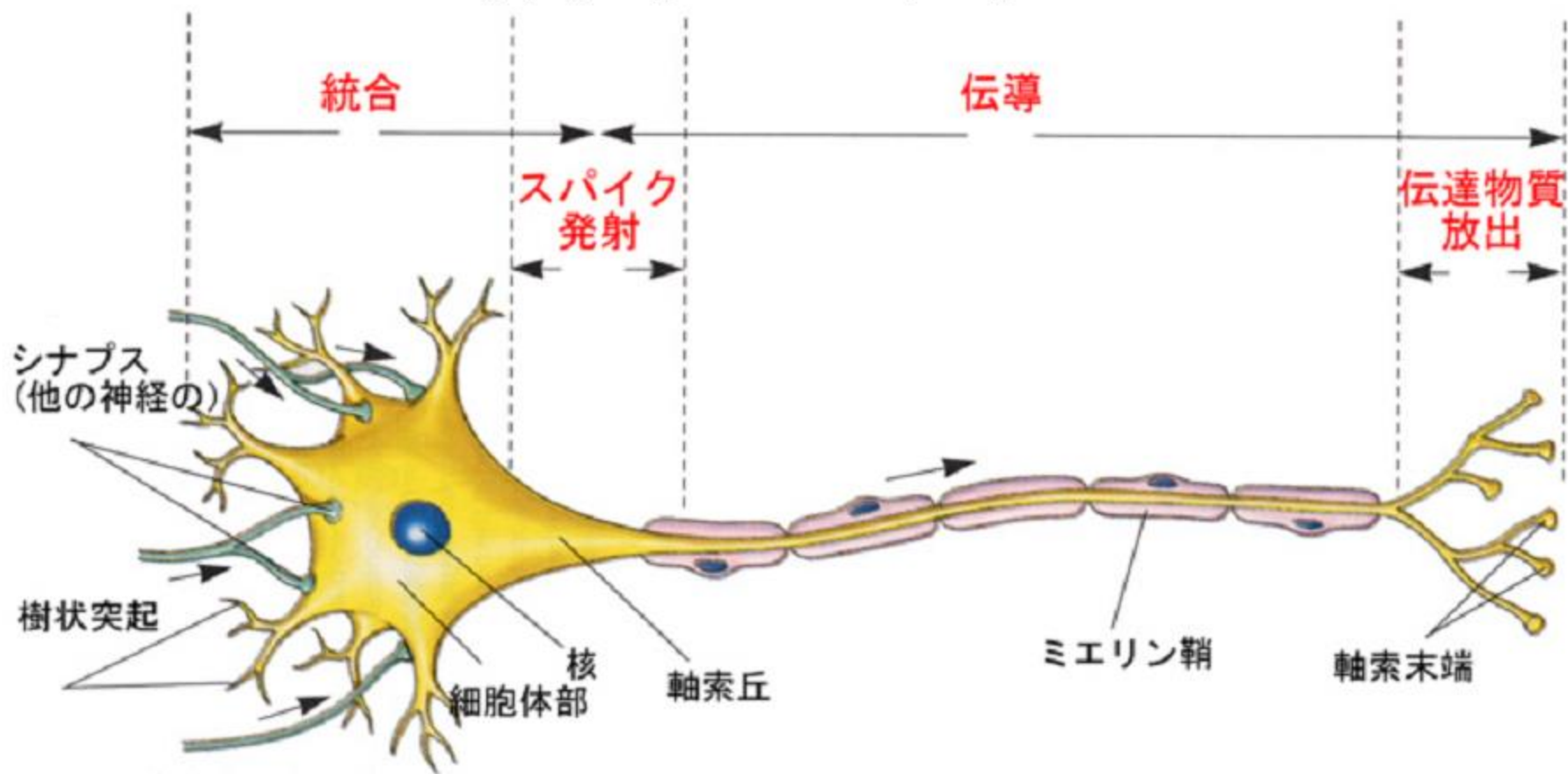




シナプスの働き



神経細胞の仕組み



別の神経細胞からシナプスを通して神経伝達物質を受け取る。
刺激が強いと神経細胞は興奮し、スパイクという電気を流す。

<http://kitsuon-kaizen.en.que.jp/hori/108.htm>

シナプス可塑性 -- ヘッブの法則

ヘッブの法則は、**脳のシナプス可塑性**についての法則である。ヘッブ則、ヘブ則とも呼ばれる。心理学者のDonald O. Hebbによって提唱された。

ニューロン間の接合部であるシナプスにおいて、シナプス前ニューロンの繰り返し発火によってシナプス後ニューロンに発火が起こると、そのシナプスの**伝達効率が**増強される。

また逆に、発火が長期間起こらないと、そのシナプスの**伝達効率は減退**するというものである。

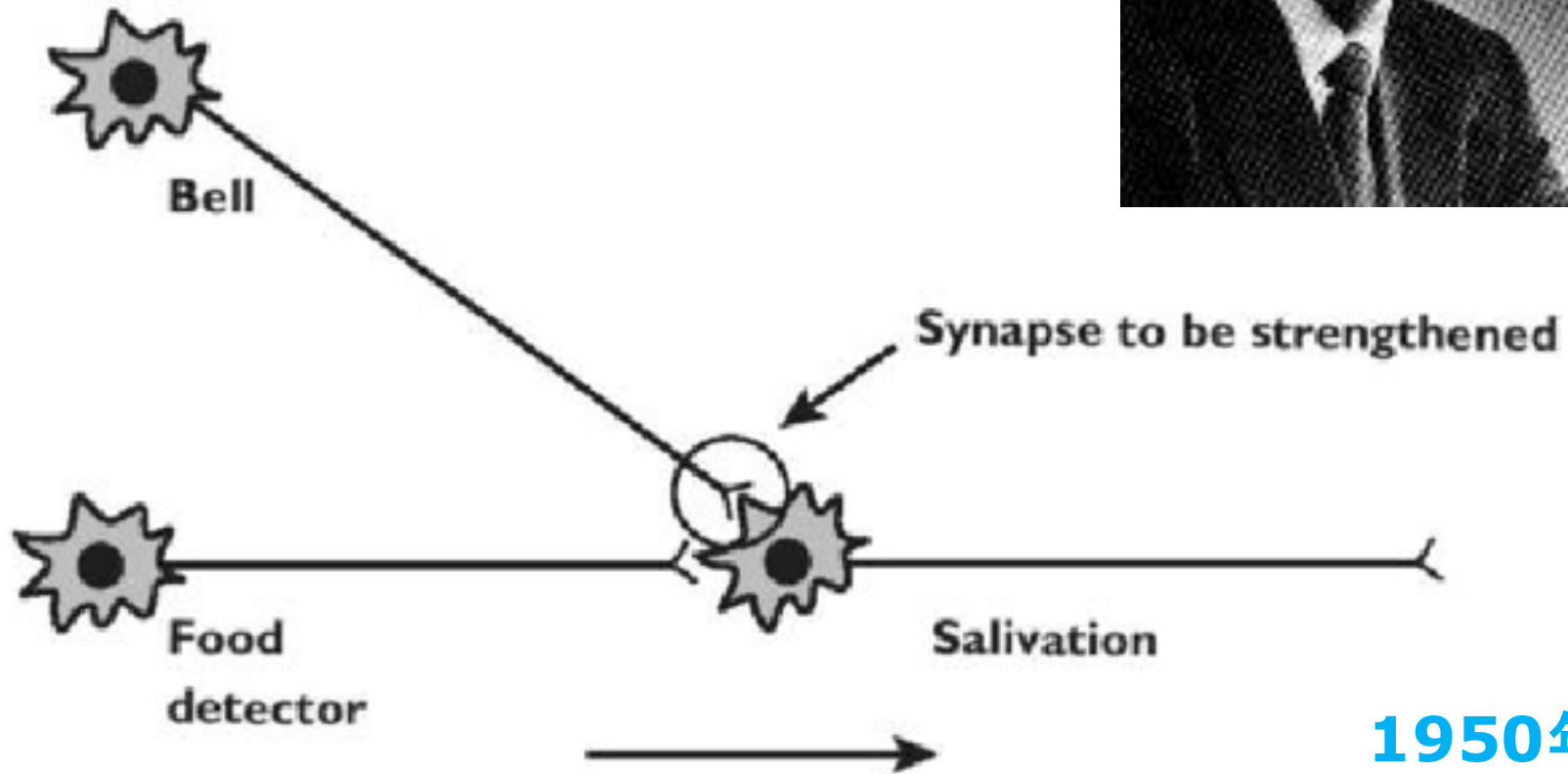
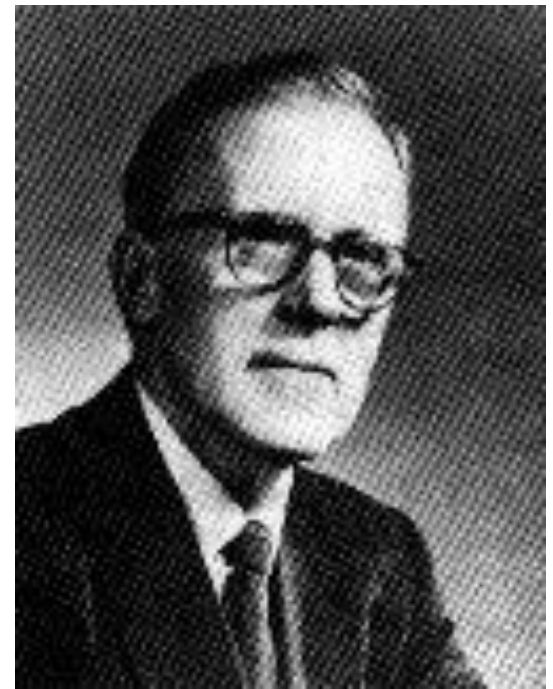
The Organization of Behavior. 1949年

<https://goo.gl/2HsDwK>



ヘッブの法則

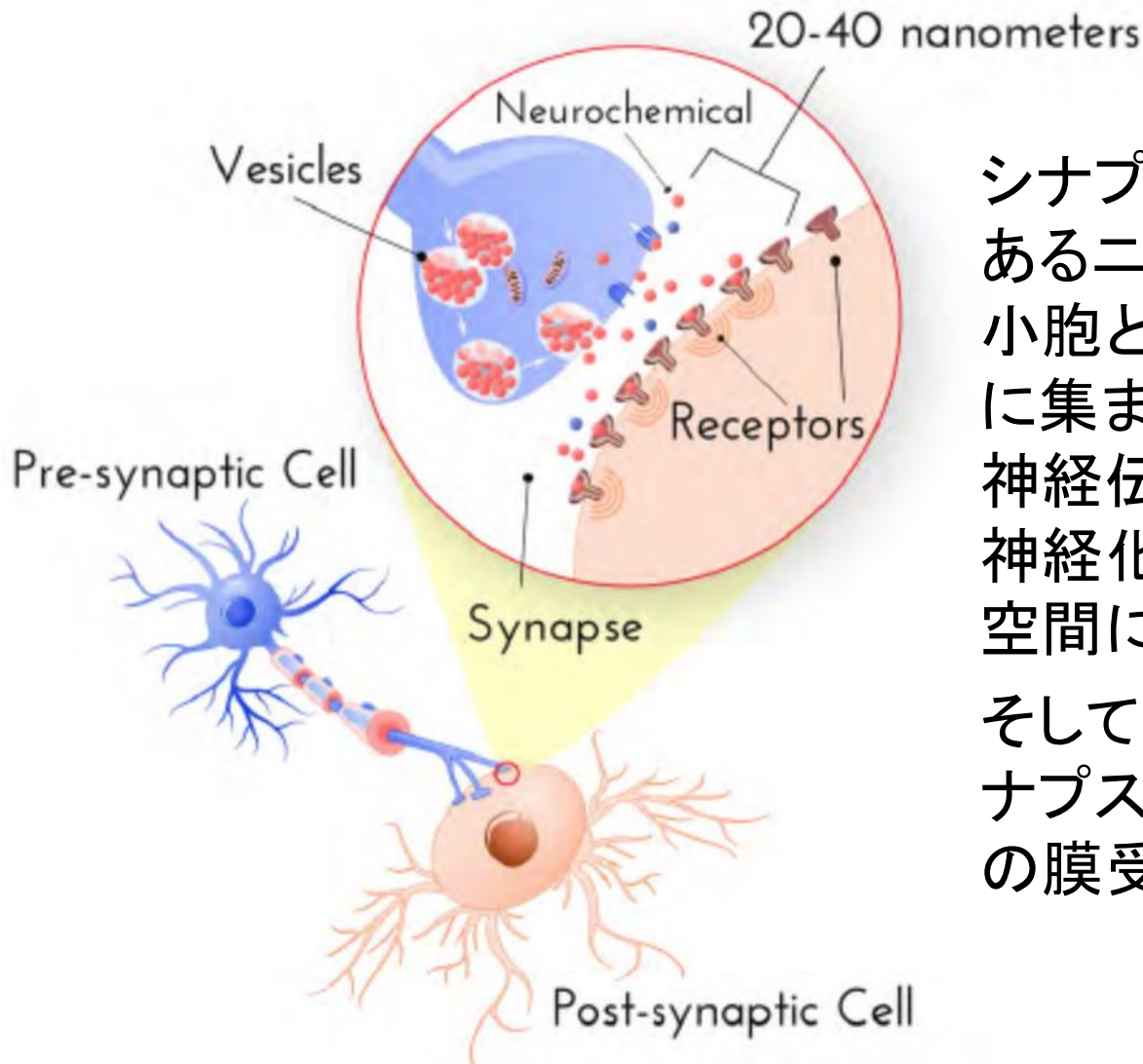
Donald O. Hebb



1950年代



シナプス伝達



シナプス前細胞と呼ばれるあるニューロンは、シナプス小胞と呼ばれる細胞膜付近に集まった特殊な袋から、神経伝達物質やその他の神経化学物質を細胞間の空間に放出する。

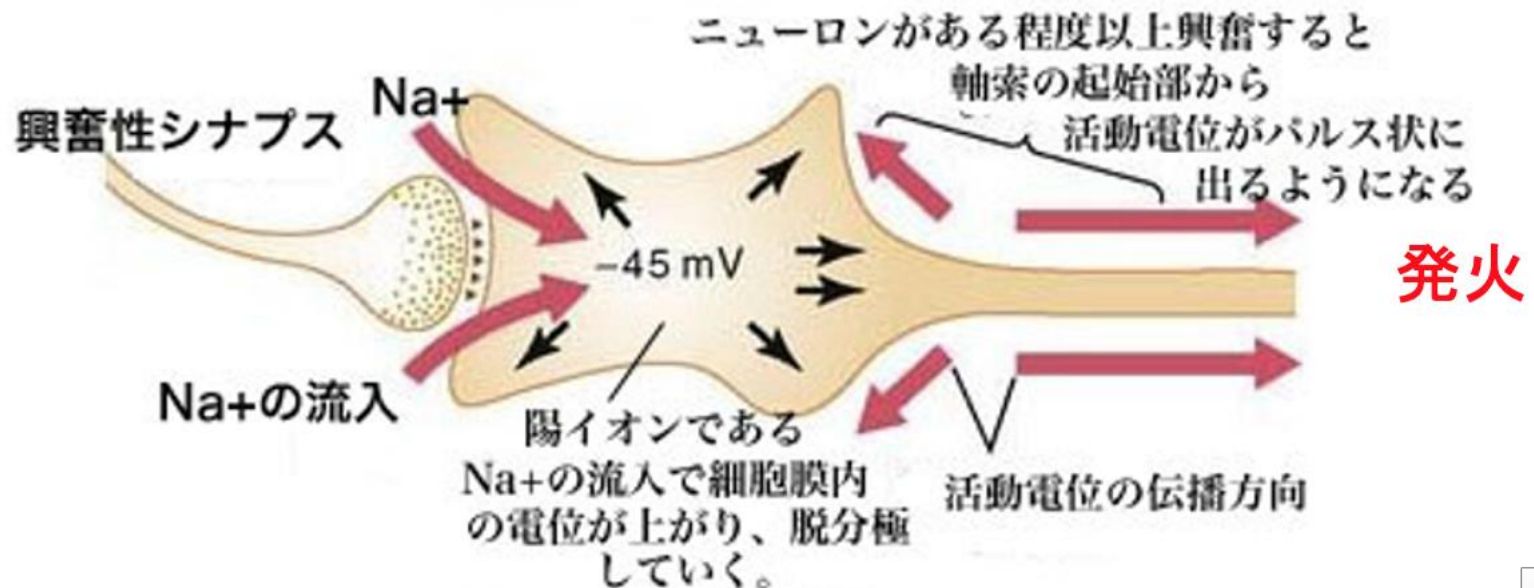
そしてそれらの分子は、シナプス後、つまり隣の細胞の膜受容体に取り込まれる。



シナプス伝達

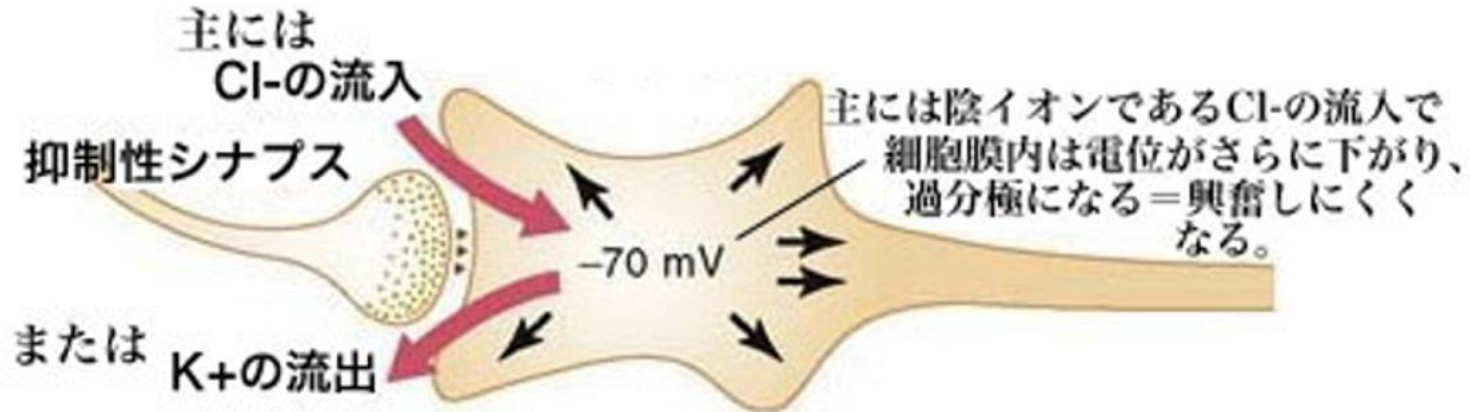
このメッセージがシナプスで2つの細胞間に伝達されると、両方の細胞の行動を変化させる力を持つ。

シナプス前ニューロンからの化学物質はシナプス後細胞を興奮させ、その細胞自身の神経化学物質を放出するように指示するかもしれない。(興奮性シナプス)



シナプス伝達

シナプス後細胞に対して、シグナル伝達を遅くするように、あるいはシグナル伝達を完全に止めるように指示することもある。(抑制性シナプス)



あるいは、単にメッセージを少し変えるように指示するかもしれない。しかしシナプスは双方向コミュニケーションの可能性を提供する。そのため、シナプス後細胞はシナプス前細胞に、神経伝達物質の放出量や放出頻度を変えるよう、自分自身のメッセージを送り返すことができる。





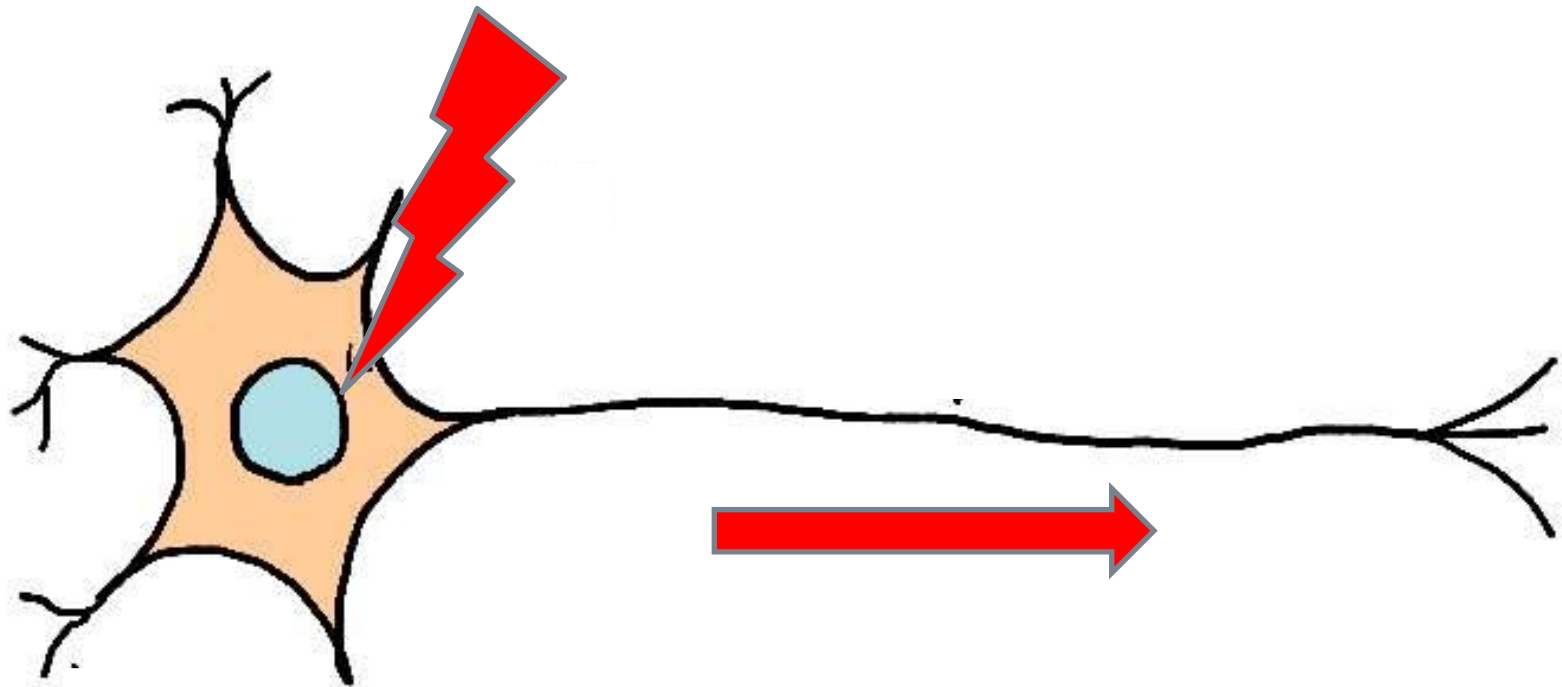
ニューロンは、いつ発火するのか？



ニューロンは、いつ発火するのか？



ニューロンは、
いつ発火するのか？



ニューロンは、 いつ発火するのか？

抑制シナプス

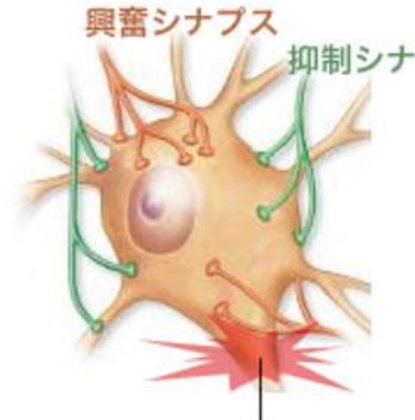
興奮シナプス

抑制シナプス

興奮シナプス

興奮シナプス

抑制シナプス



ニューロンの働きは、 発火するか発火しないかの 1か0かのデジタル・スタイル

抑制シナプス

興奮シナプス

抑制シナプス

興奮シナプス

興奮シナプス

抑制シナプス

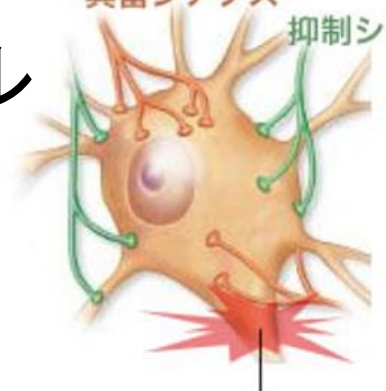
興奮シナプス

抑制シナ

興奮性シナプスからの信号の強さから
抑制性シナプスからの信号の強さを
引いた値がある閾値を越えると、
ニューロンは、「発火」する。

個々のシナプスの性質

同じ発火信号を受けても、受け止めるニューロン内
でのその信号の「強さ」は、シナプスごとに異なる。



ニューロンの働きは、 発火するか発火しないかの 1か0かのデジタル・スタイル

抑制シナプス

興奮シナプス

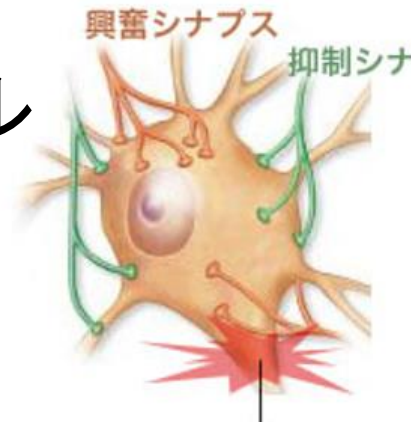
抑制シナプス

抑制シナプス

興奮性シナプスからの信号の強さから
抑制性シナプスからの信号の強さを
引いた値がある閾値を越えると、
ニューロンは、「発火」する。

個々のシナプスの性質

同じ発火信号を受けても、受け止めるニューロン内
でのその信号の「強さ」は、シナプスごとに異なる。



興奮シナプス

興奮シナプス

抑制シナプス



ニューロン発火の条件

- あるニューロンが「発火」するかは、次のようにして決まる。

興奮性シナプスから受け取る信号の強さ全体 A から
抑制性シナプスから受け取る信号の強さ全体 B を
引いて、その値がある閾値 C より大きければ発火する。

$$A - B > C \quad \text{発火} \quad (A - B - C > 0)$$

$$A - B < C \quad \text{発火しない} \quad (A - B - C < 0)$$

- 発火賛成と発火反対の多数決。ただし、賛成票が、ある一定数以上（閾値）、反対票を上回らないといけないというルール。至極、単純である。

$$\text{賛成票} - \text{反対票} > \text{閾値} \quad \text{発火}$$

$$\text{賛成票} - \text{反対票} < \text{閾値} \quad \text{発火しない}$$



ニューロン発火の判断

賛成票 - 反対票 > 閾値 発火

X1=1

X2=1

X3=1

X4=1

X5=1

X6=1

閾値 = 1 の場合

賛成票: $3+3+2=7$

反対票: $2+1+2=5$

賛成票 - 反対票 = $2 > 1$ (閾値)

発火!

個々のシナプスの性質は異なる

内部の数字は、シナプスごとに異なる
信号を受けた時に伝えられる、信号の「強さ」

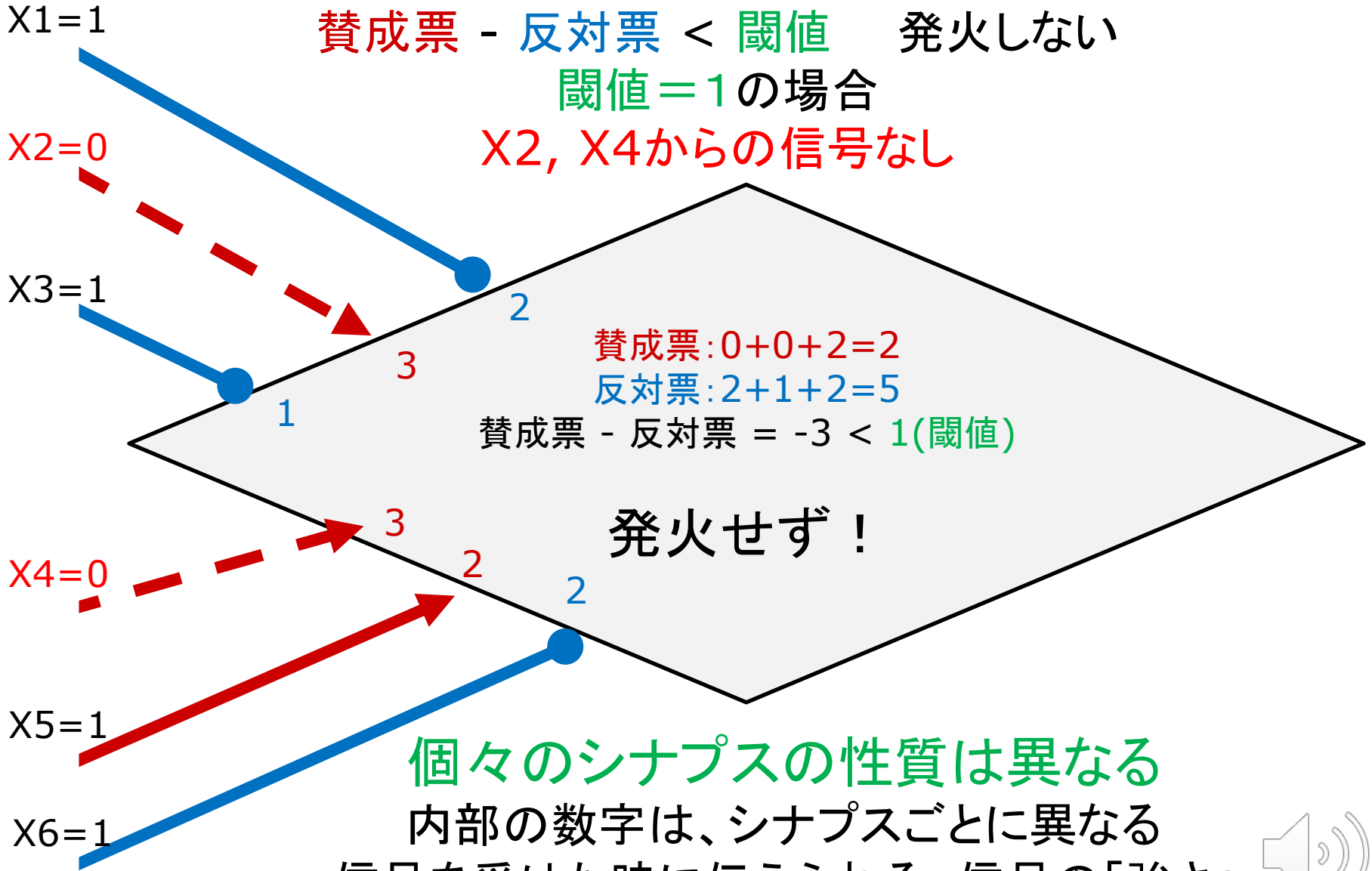


ニューロン発火の判断

賛成票 - 反対票 < 閾値 発火しない

閾値 = 1 の場合

X2, X4からの信号なし



賛成票: $0+0+2=2$

反対票: $2+1+2=5$

賛成票 - 反対票 = $-3 < 1$ (閾値)

発火せず!

個々のシナプスの性質は異なる

内部の数字は、シナプスごとに異なる
信号を受けた時に伝えられる、信号の「強さ」



ニューロン発火の判断

賛成票 - 反対票 > 閾値 発火

閾値 = 1 の場合

X3, X5からの信号なし

X1=1

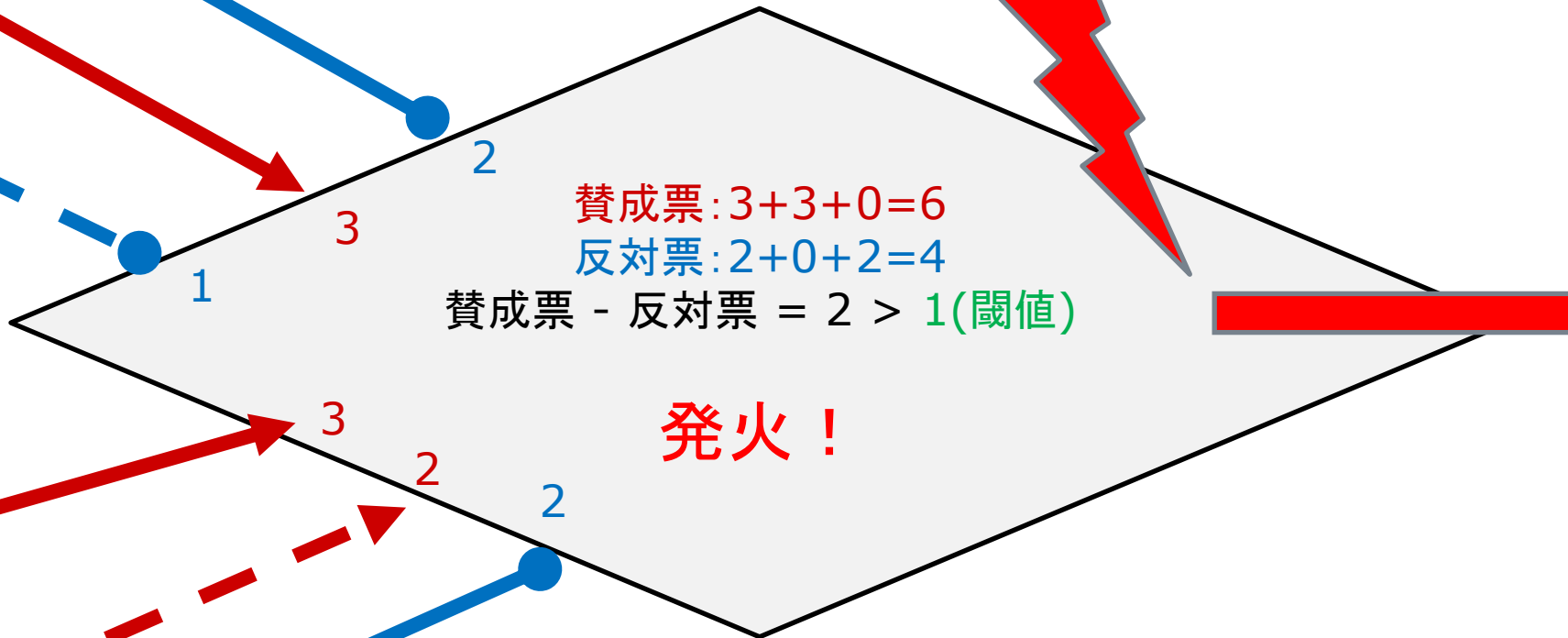
X2=1

X3=0

X4=1

X5=0

X6=1



個々のシナプスの性質は異なる

内部の数字は、シナプスごとに異なる
信号を受けた時に伝えられる、信号の「強さ」



「重み」と「バイアス」

- 賛成票の数 A をプラスの値に
反対票の数 B をマイナスの値 B' にし、
閾値 C の符号を逆にして C' (バイアス)とすると、
発火の条件 $A - B > C$ (すなわち、 $A - B - C > 0$)は、
 $A + B' + C' > 0$ の形になる。
- 各シナプスごとに異なる投票権の大きさを「重み」という。
興奮性シナプスはプラスの「重み」を持ち、抑圧性シナプスはマイナスの「重み」を持つと考えるということ。
- 一つの入力 X_i (0か1の値をとる)には、それに対応した一つのシナプスが存在する。そのシナプスの重みを W_i とすれば、この一つのシナプスの投票数は、入力 X_i に 重み W_i をかけたものになる。



入力 X_i と重み W_i とバイアス b で
発火の条件を表す

$$\sum_i W_i X_i + b > 0$$



入力 X_i と重み W_i とバイアス b で 発火の条件を表す

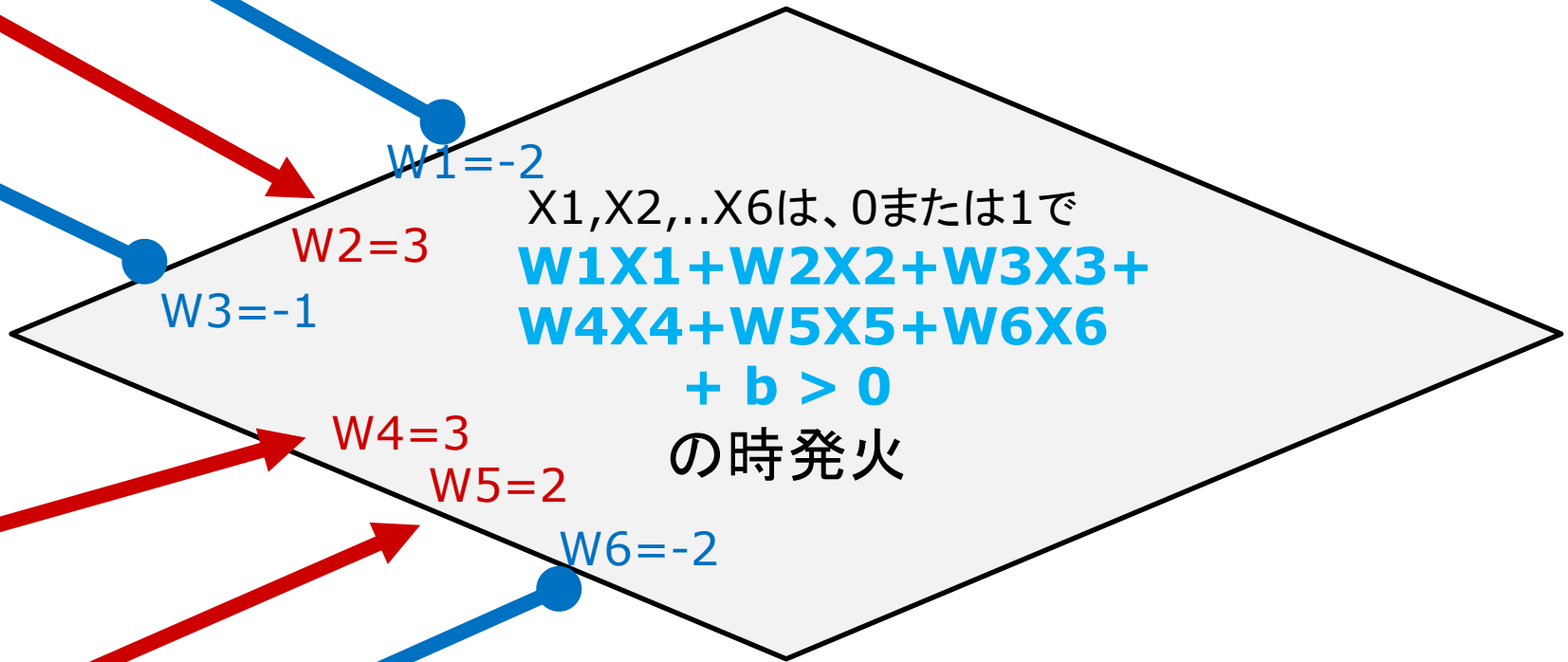
- 今、6つの入力 X_i ($i=1,2,\dots,6$)があったとしよう。この時、この入力に対応するシナプスの重みを W_i ($i=1,2,\dots,6$)とすれば、各シナプスでの投票数は、入力 X_i に対応する重み W_i をかけたものだから、一つのニューロン全体の投票数は、このニューロンのすべての6個のシナプスの投票を足し合わせたものになる(符号も込みで考えている)。
- この投票数にバイアスを足し合わせたものがプラスの時、このニューロンは発火する。
- 式で表すと、入力 X_1, X_2, \dots, X_6 は、0または1で
 $W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4 + W_5X_5 + W_6X_6 + b > 0$ の時発火ということになる。



重み: W_i

シナプスごとの賛成票の投票権を**プラスの値**に
反対票の投票権を**マイナスの値**にすると、
式が簡単になる。

X1
X2
X3
X4
X5
X6



X1, X2, ..., X6は、0または1で
 $W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4 + W_5X_5 + W_6X_6 + b > 0$

の時発火

バイアス: b

ついでに、**閾値 C** の符号を逆にすれば、
式はもっと簡単になる。それを「バイアス」という。



重み $W_i = [-2, 3, -1, 3, 2, -2]$

バイアス $b = -1$

入力 $X_i = [1, 0, 1, 0, 1, 1]$

の場合

$X_1 = 1$

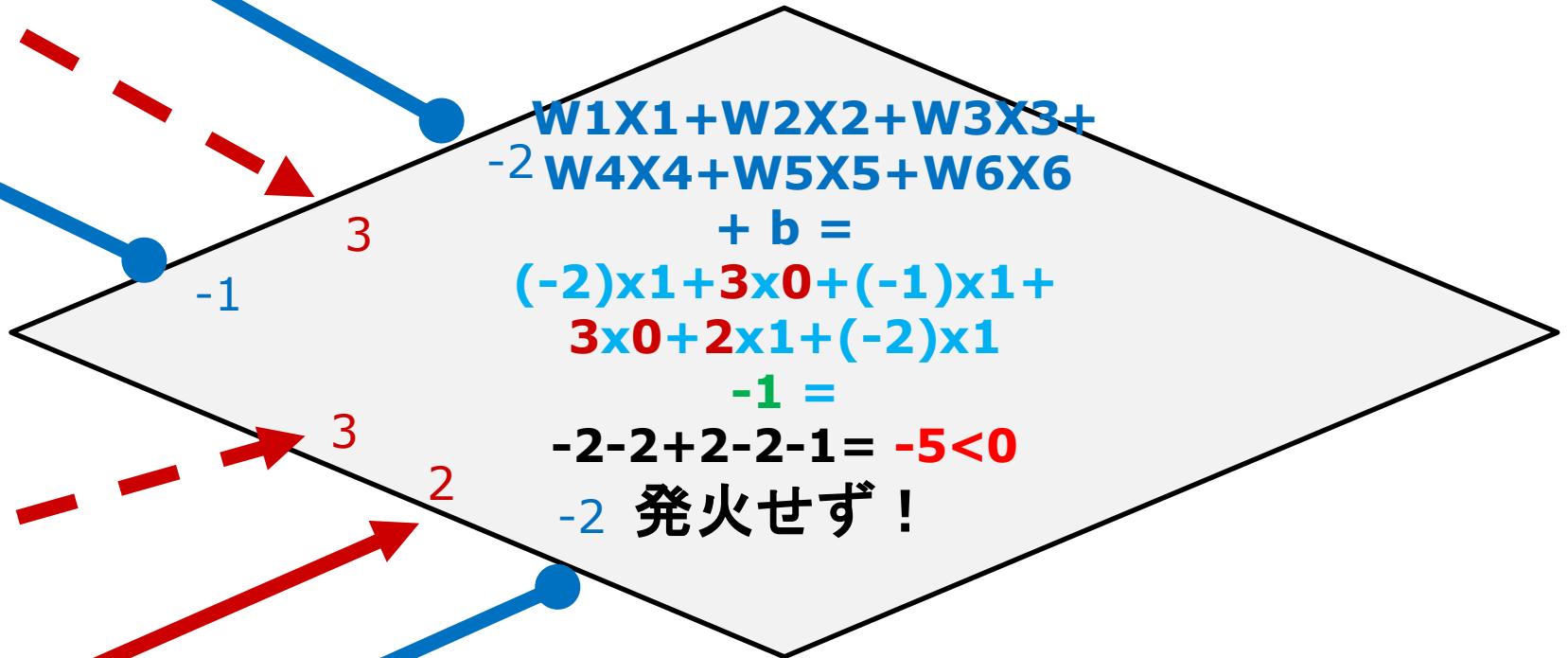
$X_2 = 0$

$X_3 = 1$

$X_4 = 0$

$X_5 = 1$

$X_6 = 1$



重み $W_i = [-2, 3, -1, 3, 2, -2]$

バイアス $b = -1$

入力 $X_i = [1, 1, 0, 1, 0, 1]$

の場合

$X_1 = 1$

$X_2 = 1$

$X_3 = 0$

$X_4 = 1$

$X_5 = 0$

$X_6 = 1$

$$W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4 + W_5X_5 + W_6X_6$$

+ b =

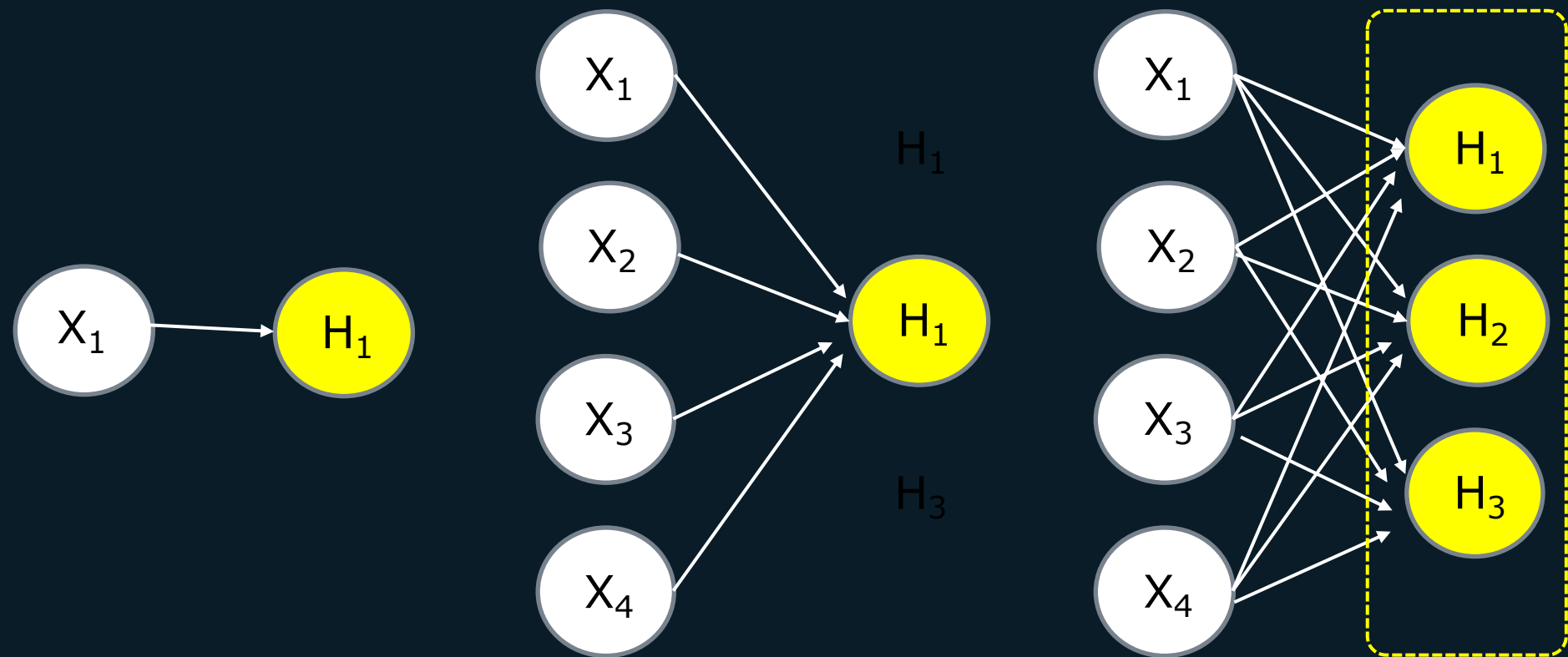
$$(-2) \times 1 + 3 \times 1 + (-1) \times 0 + 3 \times 1 + 2 \times 0 + (-2) \times 1$$

-1 =

$$-2 + 3 + 3 - 2 = 2 > 0$$

発火!





ニューラル・ネットワークの
状態を式で表現する



生物のニューロンから コンピュータ上のニューロンへ

新しい表記



重み $W_i = [-2, 3, -1, 3, 2, -2]$

バイアス $b = -1$

入力 $X_i = [1, 0, 1, 0, 1, 1]$

の場合

$X_1 = 1$

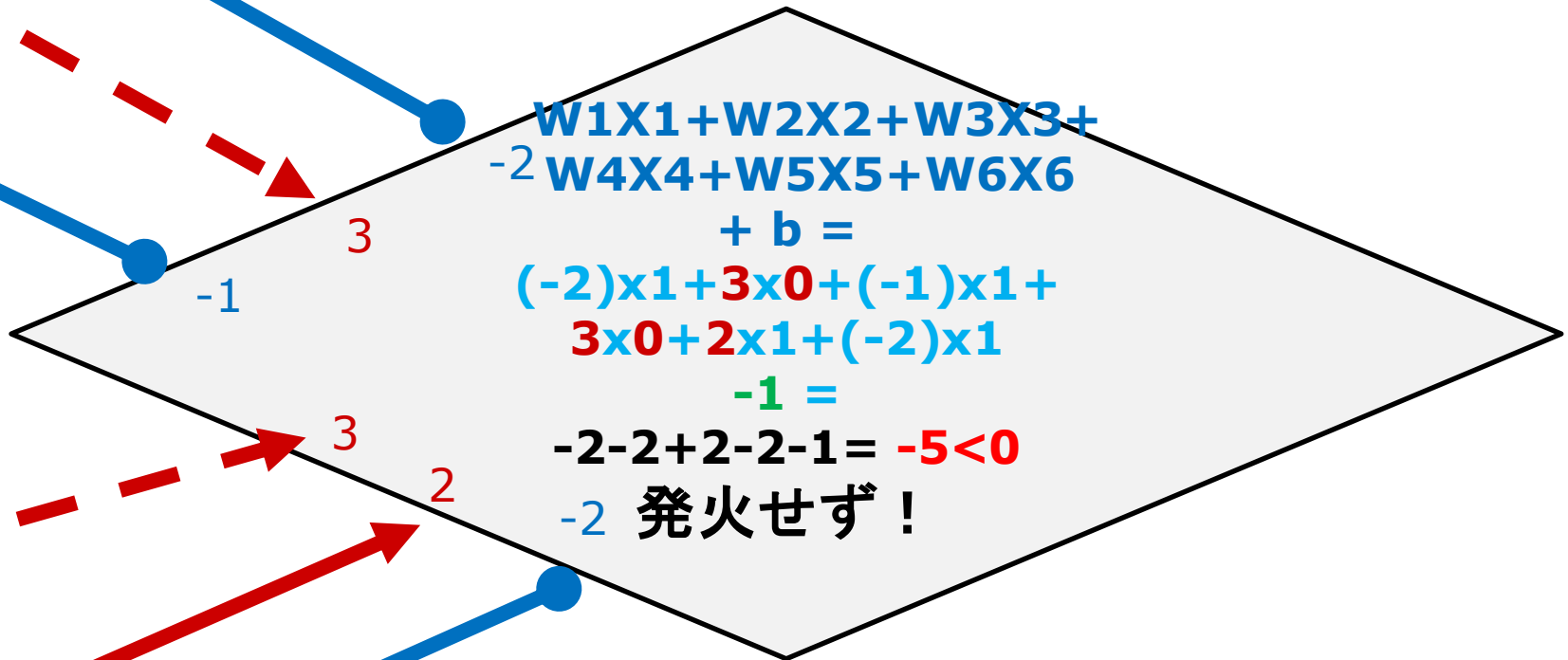
$X_2 = 0$

$X_3 = 1$

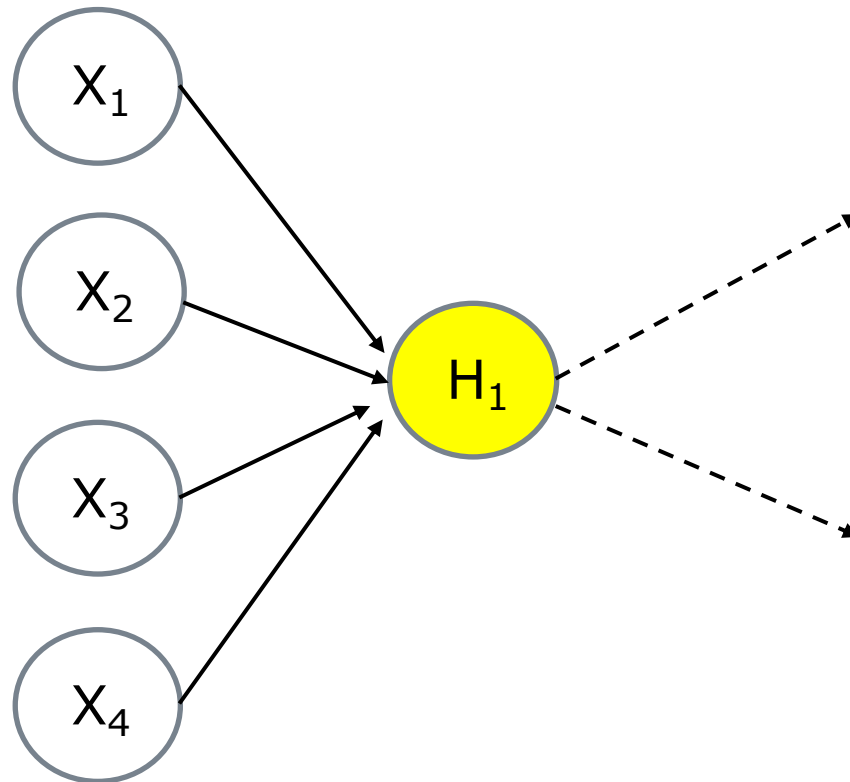
$X_4 = 0$

$X_5 = 1$

$X_6 = 1$



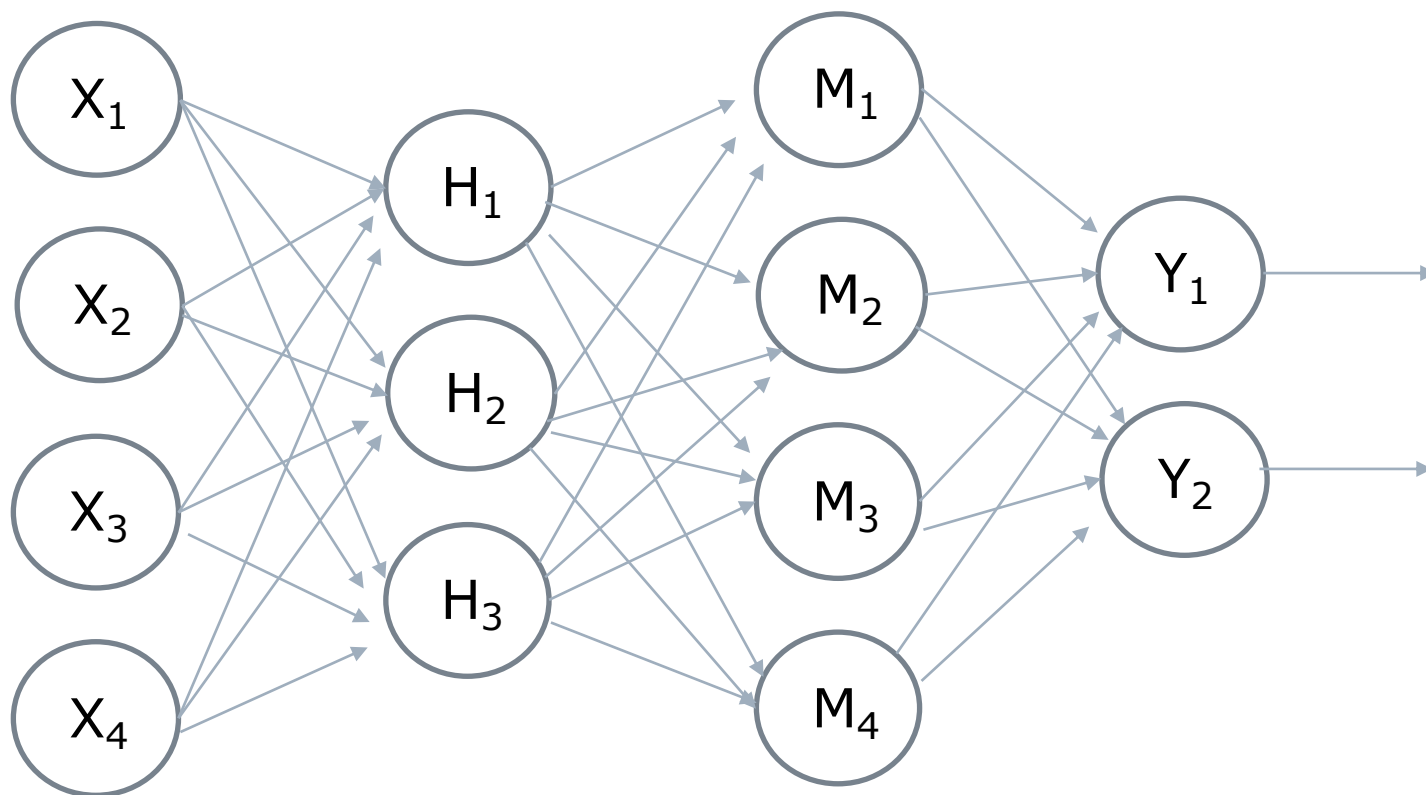
ニューラル・ネットワークの新しい表記



一つの丸が、一個のニューロンを表している。
一つのニューロンは、その「出力」を、他のニューロンに、送り出す。



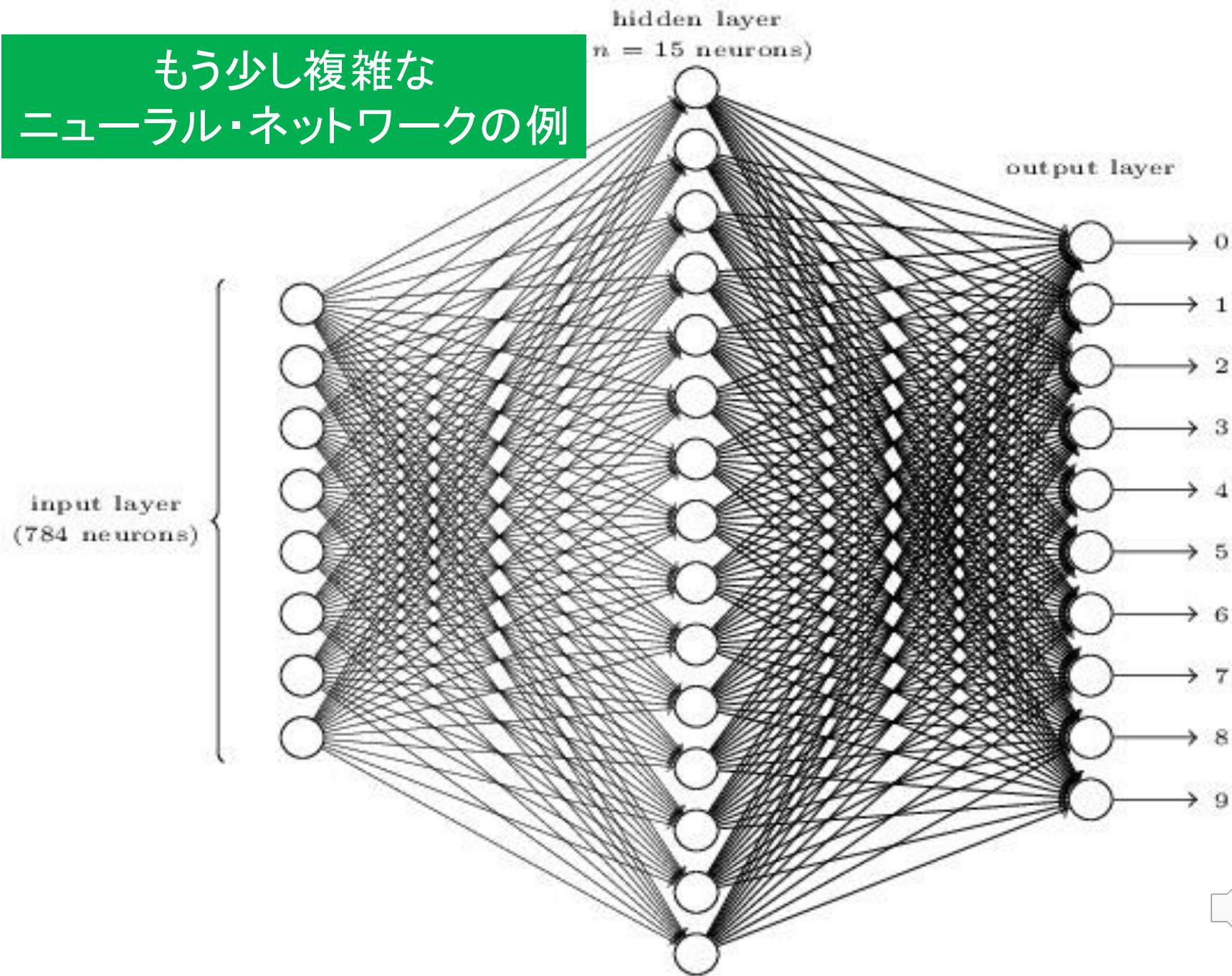
ニューラル・ネットワークの表記の例



一つ一つの丸が、一個のニューロンを表している。
一つのニューロンは、その「出力」を、他のニューロンに、送り出す。



もう少し複雑な
ニューラル・ネットワークの例





ニューラル・ネットワークの状態の表現
単純なものから複雑なものへ



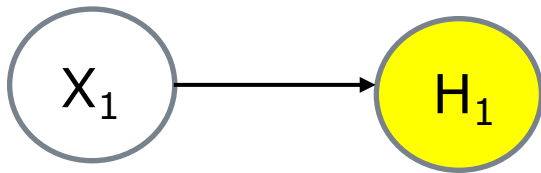
一つのニューロンの 内部の状態の表現

一つのニューロンの内部の状態は、ニューロンがシナプスを通じて受け取る刺激である「入力」と、シナプスごとの「重み」と、発火の閾値に対応する「バイアス」の三つの量で表現できる。



一つの入力を受け取る 一つのニューロンの状態

ニューロンは一つのシナプスを持つ。



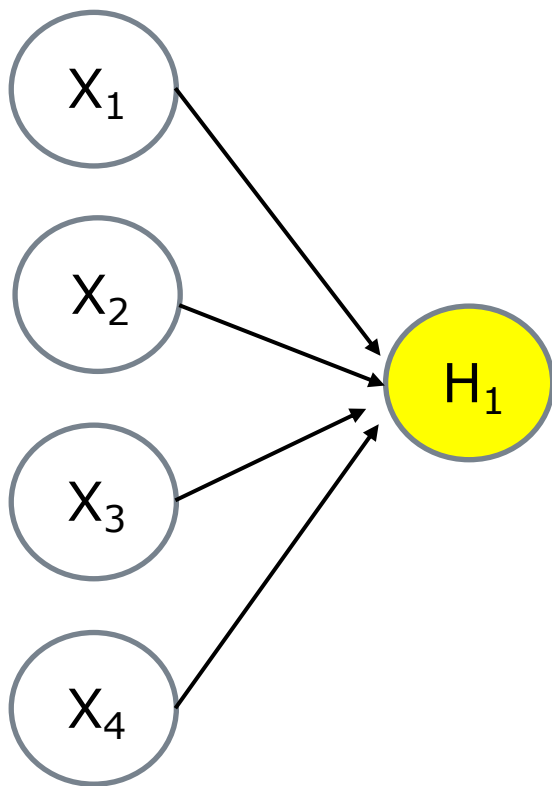
このシナプスの重みを W_1 ,
このニューロンのバイアスを b_1
とすれば、

このニューロンの状態 Y は

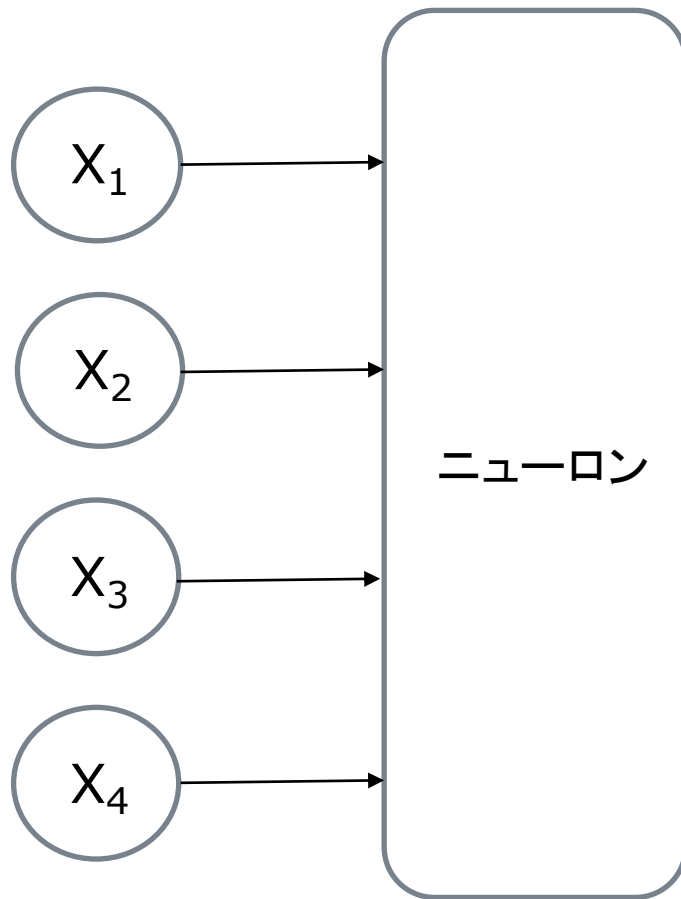
$$Y = W_1 \cdot X_1 + b_1$$



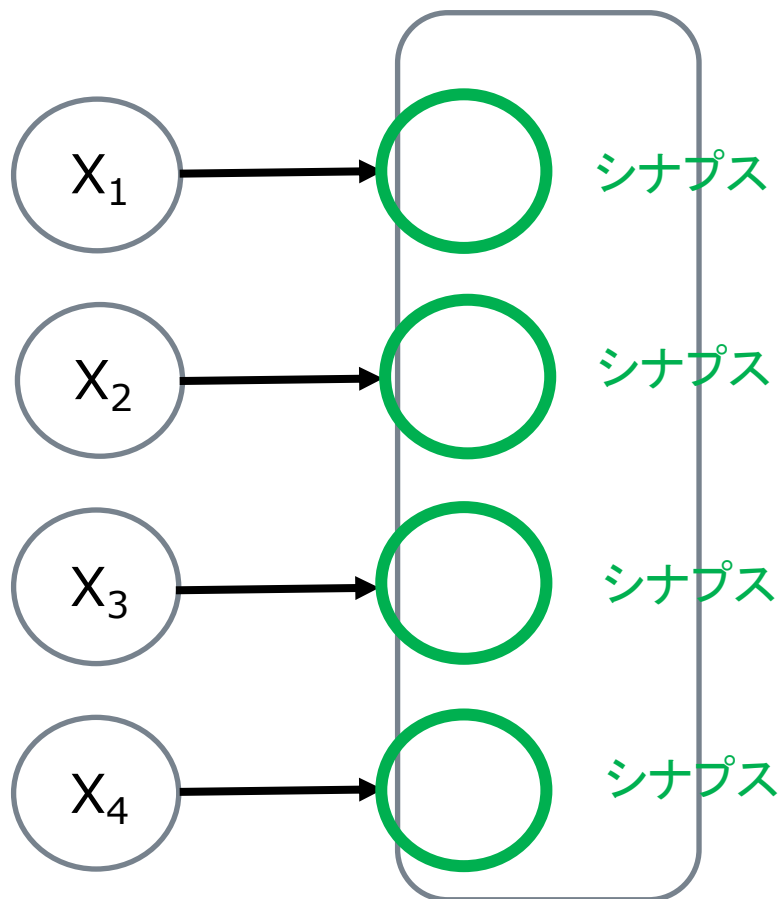
複数の入力を受け取る 一つのニューロン



複数の入力を受け取る一つのニューロン



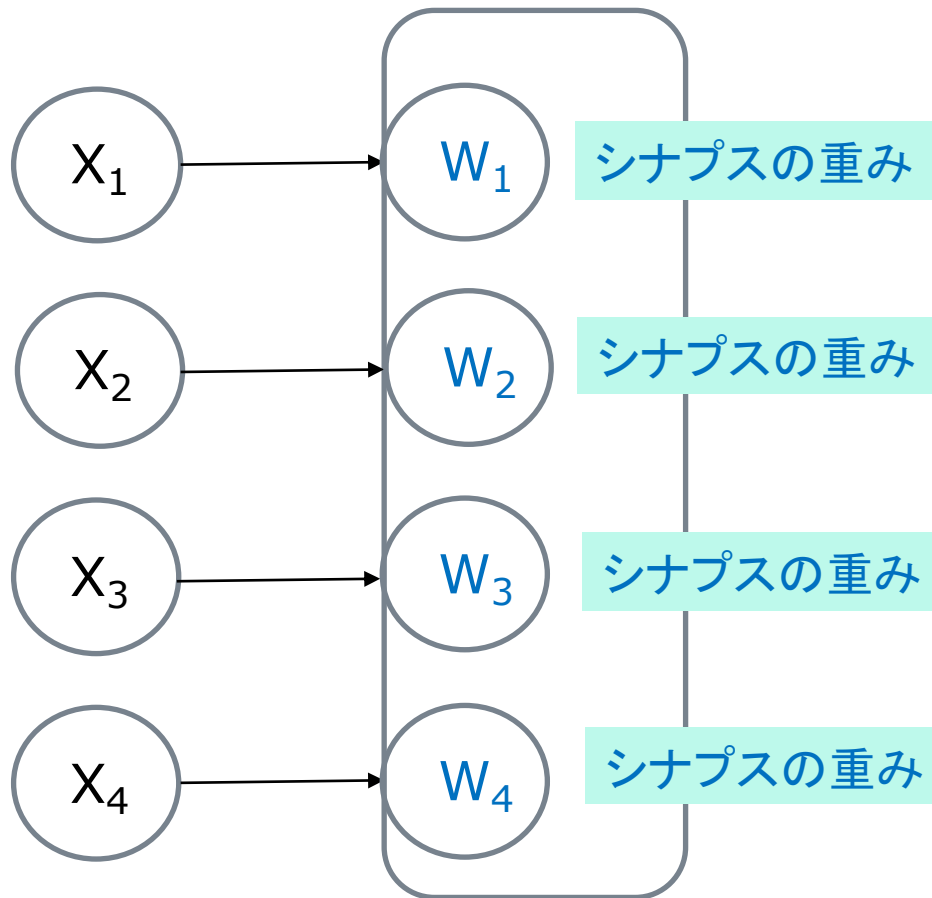
ニューロンは、シナプスを通じて入力を受け取る



シナプスの数は、
入力の数に等しい



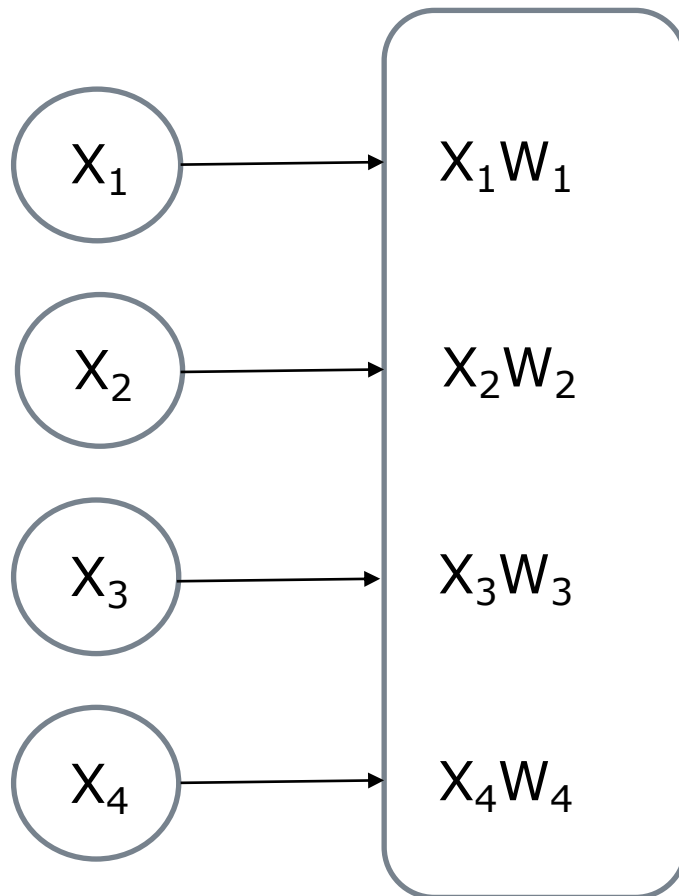
シナプスには、「**重み**」がある



重みの数は、
シナプスの数、
すなわち
入力の数に等しい



「多数決」で決めよう。入力は、シナプスの重みを掛けられてニューロンに送られる



ニューロンが出力をするか否かは、シナプスの「多数決」で決められる。

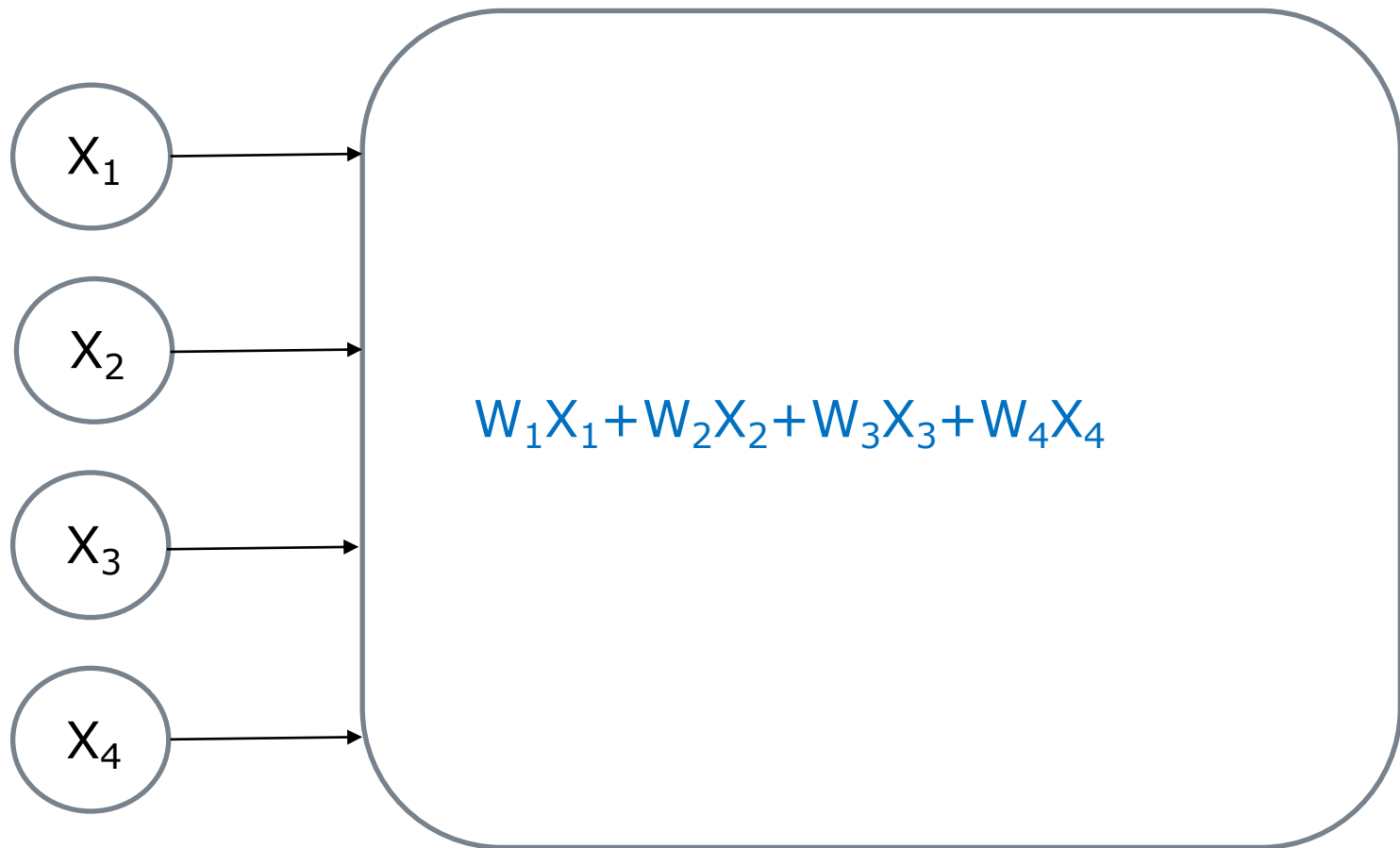
ただし、すべてのシナプスが、平等な一票を持つわけではない。シナプスは、その「重み」に応じた投票権を持つ。株主総会で、持ち株数に応じた投票権を持つのと同じである。

シナプス i が、重み W_i を持つ時、入力 X_i に対して、重みを掛けて、 $X_i \cdot W_i$ の投票を行う。

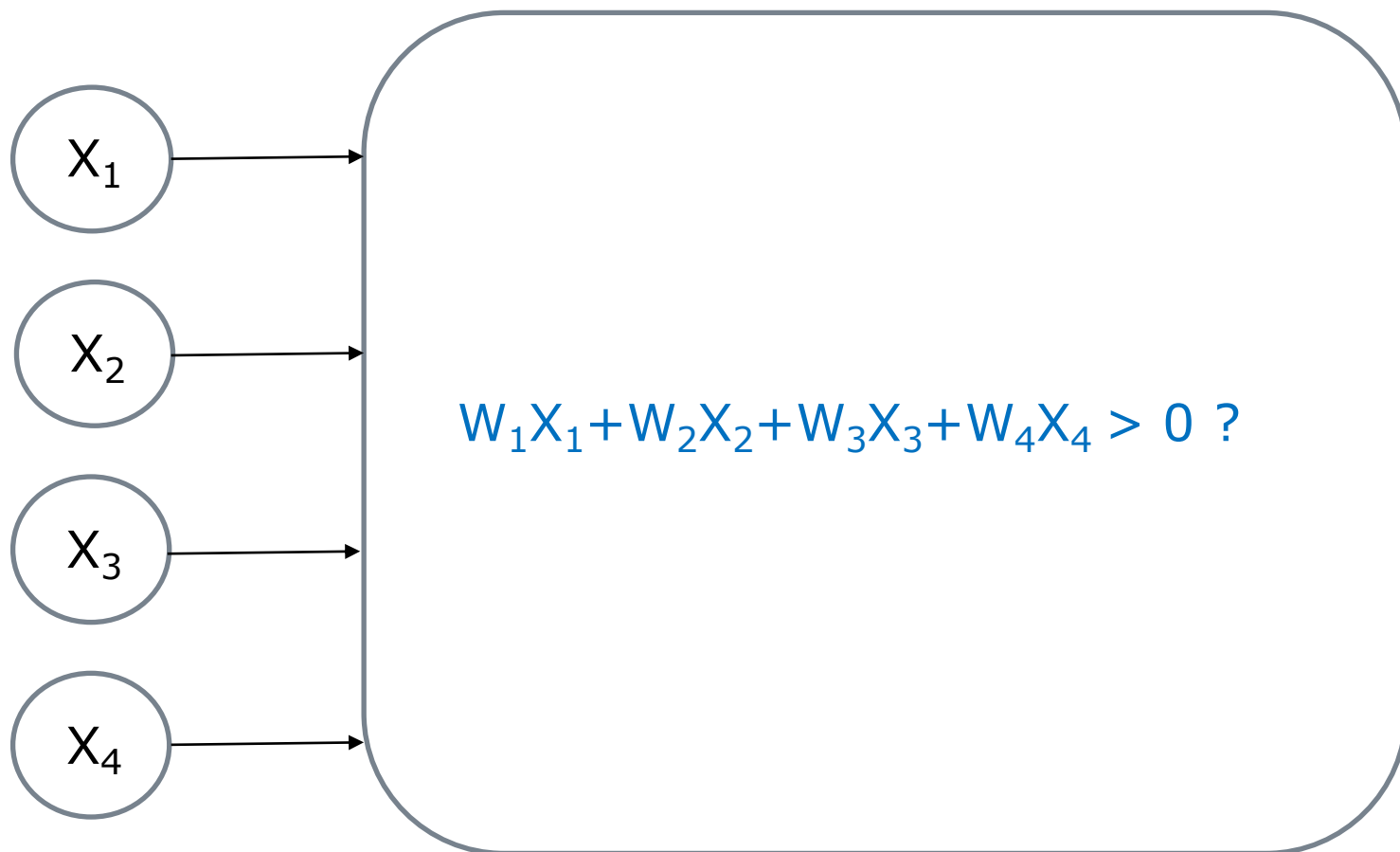
プラスの重みを出力賛成
マイナスの重みを出力反対
と、考えよう。



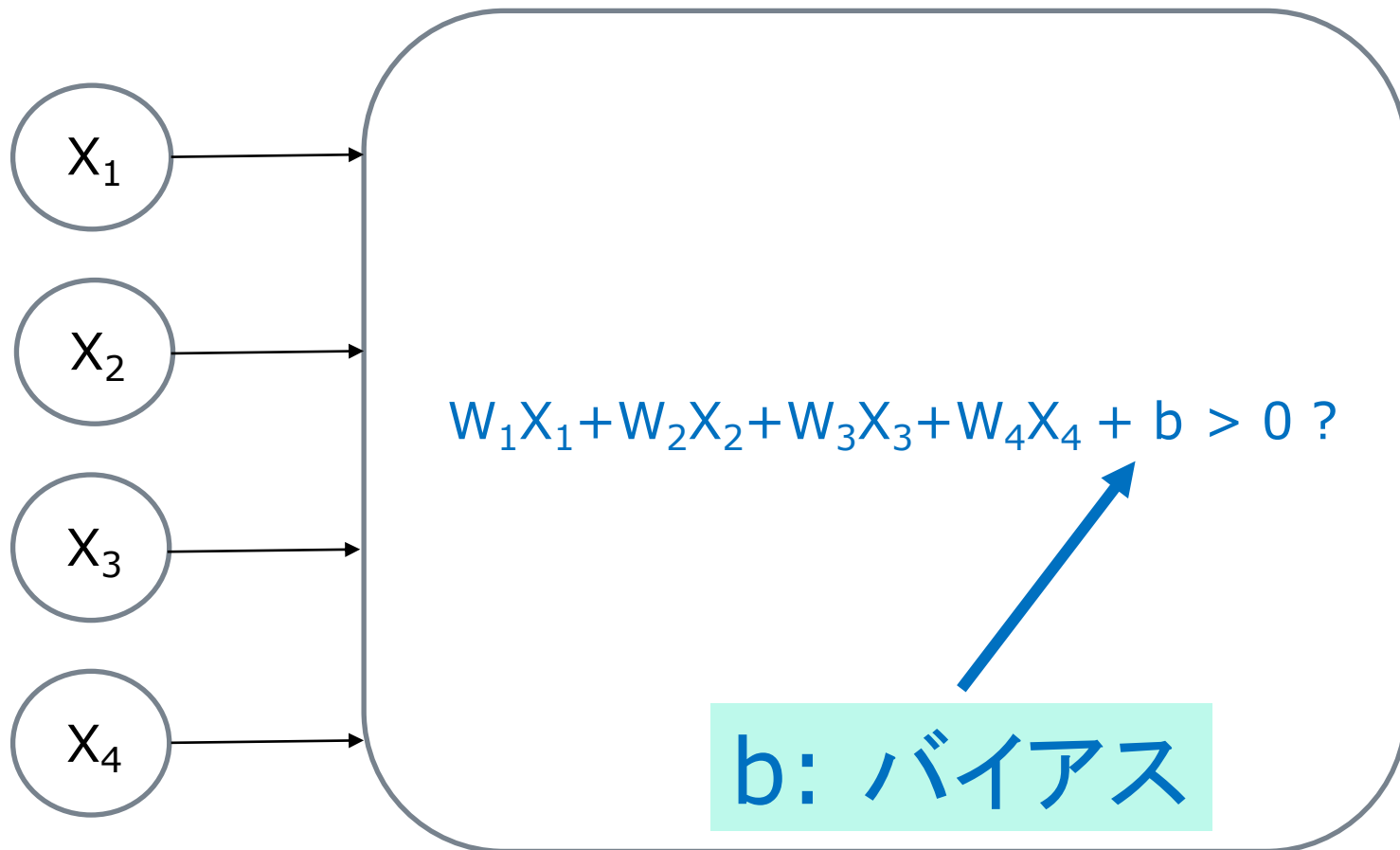
重みづけられた入力、すなわち投票を、
すべて、足し合わす。これが、投票結果。



投票結果がゼロより大きかったら、ニューロンは「発火」するの？ そうではない。



「発火」の条件には、ある閾値がある。それを**バイアス**という。投票結果とバイアスを合わせたものがゼロを超えた時、ニューロンは発火する



X_1

X_2

X_3

X_4

X_5

X_6

重み $W = [W_1, W_2, W_3, W_4, W_5, W_6]$
入力 $X = [X_1, X_2, X_3, X_4, X_5, X_6]$

バイアス b

$$W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4 + W_5X_5 + W_6X_6 + b > 0 ?$$



重みと入力の対応する項目同士を
かけて、すべて足し合わせて、それに
バイアス b を足したもの。

ニューロンの状態

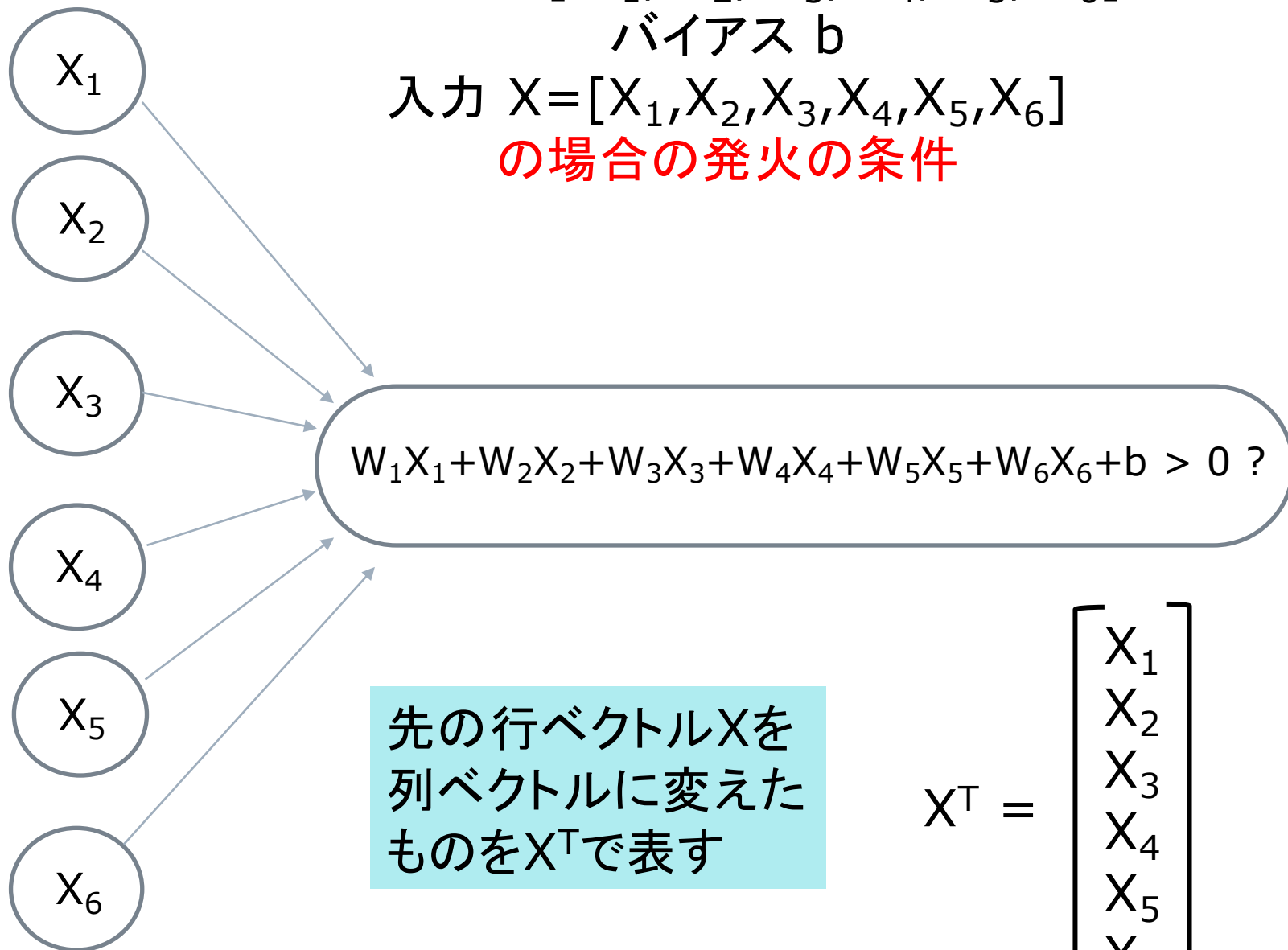


重み $W = [W_1, W_2, W_3, W_4, W_5, W_6]$

バイアス b

入力 $X = [X_1, X_2, X_3, X_4, X_5, X_6]$

の場合の発火の条件



先に行ベクトル X を
列ベクトルに変えた
ものを X^T で表す

$$X^T = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix}$$



行ベクトルと列ベクトルの積 ベクトルの内積

- 行ベクトル $W=[W_1, W_2, W_3, W_4, W_5, W_6]$ と
行ベクトル $X=[X_1, X_2, X_3, X_4, X_5, X_6]$ を列ベクトルに変えた X^T との積 $W \cdot X^T$ を次のように定義する。

- $W \cdot X^T =$

$$[W_1, W_2, W_3, W_4, W_5, W_6] \cdot \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix}$$

この積を、二つの
ベクトルの内積
という

$$= W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4 + W_5X_5 + W_6X_6$$

対応する要素を掛けて、足し合わせたものである。



この時、

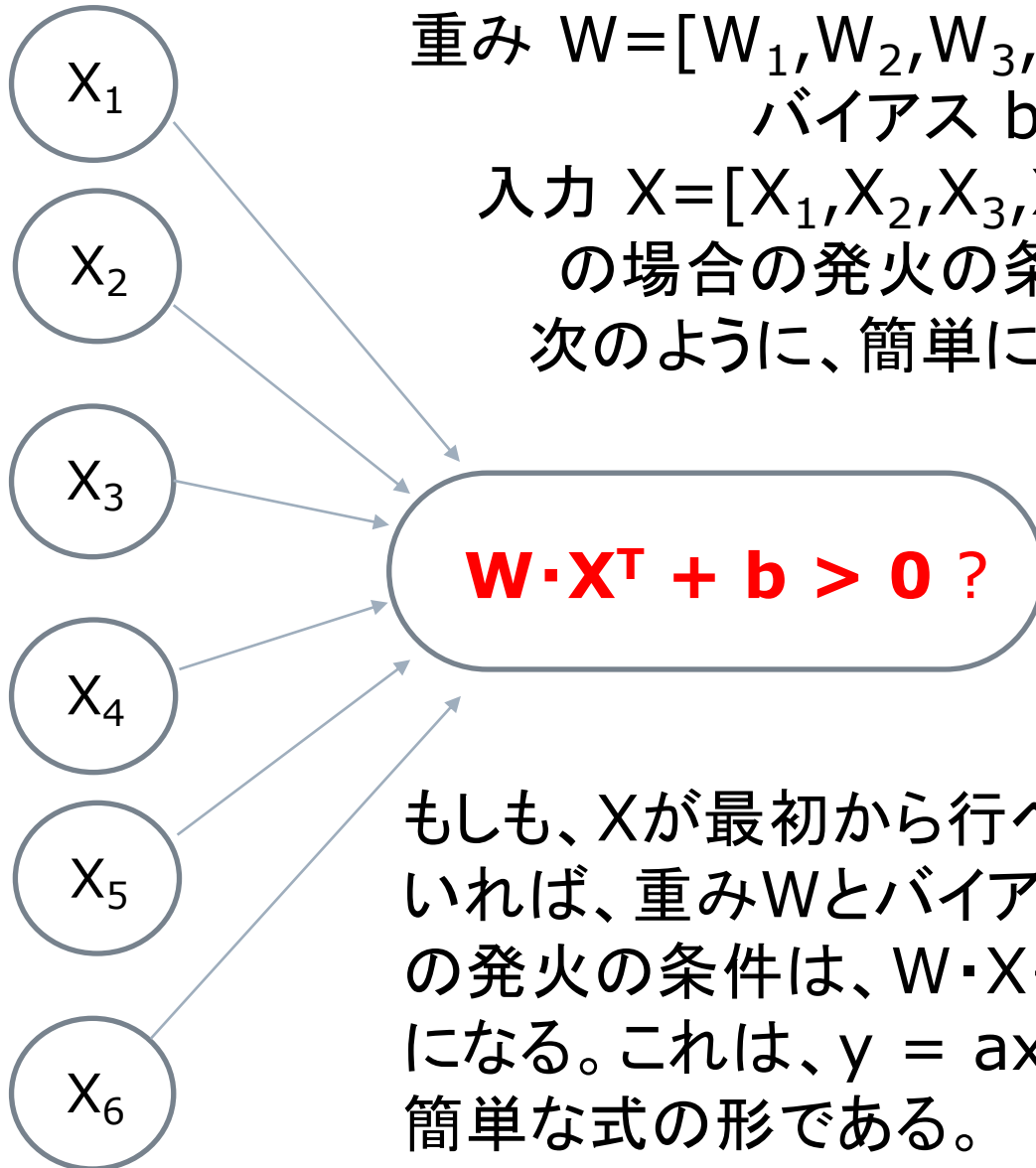
重み $W = [W_1, W_2, W_3, W_4, W_5, W_6]$

バイアス b

入力 $X = [X_1, X_2, X_3, X_4, X_5, X_6]$

の場合の発火の条件は、

次のように、簡単に書ける。



もしも、 X が最初から行ベクトルの形で与えられていけば、重み W とバイアス b を持つあるニューロンの発火の条件は、 $W \cdot X + b > 0$ で与えられることになる。これは、 $y = ax + b > 0$ と同じくらい、簡単な式の形である。



この時、

重み $W=[W_1, W_2, W_3, W_4, W_5, W_6]$

バイアス b

入力 $X=[X_1, X_2, X_3, X_4, X_5, X_6]$

の場合の発火の条件は、
次のように、簡単に書ける。


$$W \cdot X^T + b > 0 ?$$

ニューロンの状態と
発火の条件

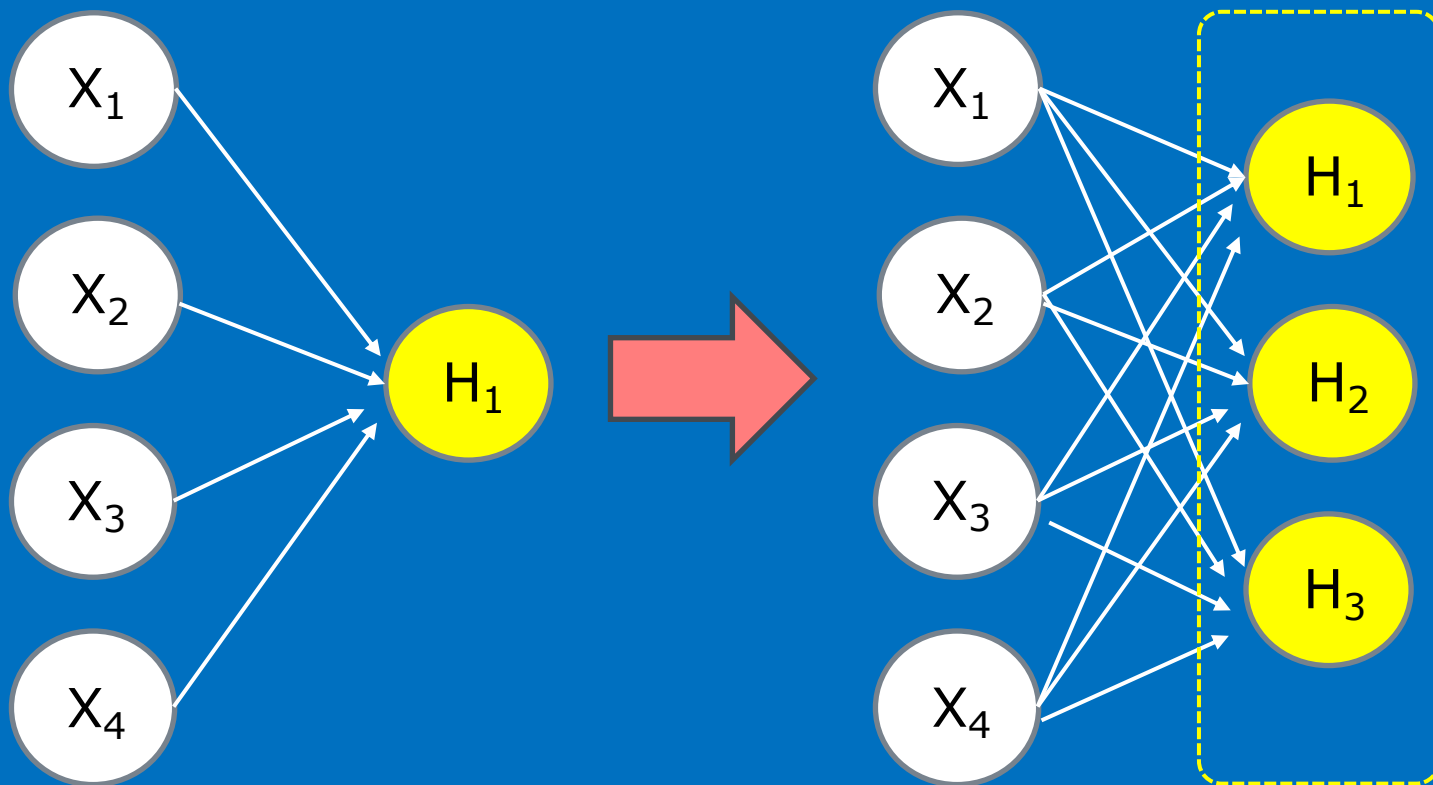
この式は、ニューラルネットワークで、
もっとも重要な式の一つである。





ニューラル・ネットワークの状態の表現
単純なものから複雑なものへ





ニューラル・ネットワークの状態の表現
単純なものから複雑なものへ



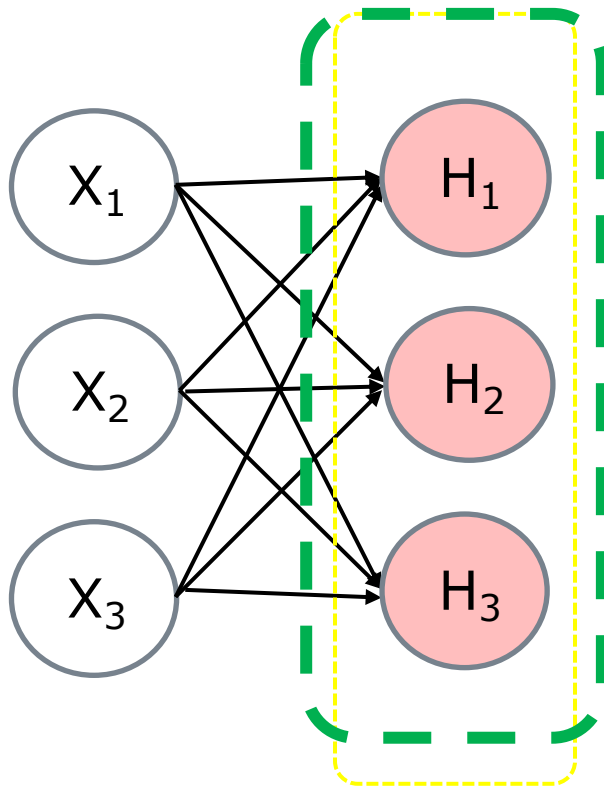
複数のニューロンからなる 一つの層の内部の状態の表現

複数のニューロンの集まりが、一つの「層」として、入力を受け取り出力を返すことがある。

この層の内部の状態も、この層が受け取る「入力の全体」と、シナプスごとの「重みの全体」と、層を構成するニューロンの閾値に対応する「バイアスの全体」という三つの量で表現できる。



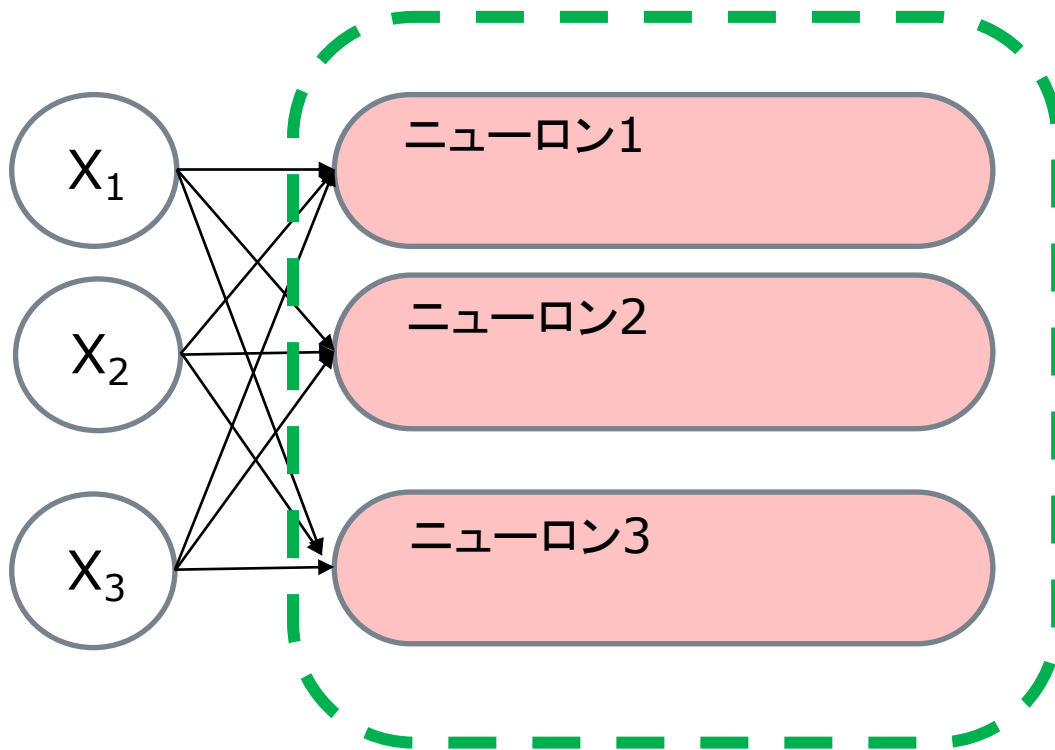
3つの入力を受け取る 3つのニューロンの状態を考える



ここでは、3つのニューロンが一つの「層」を作っている。



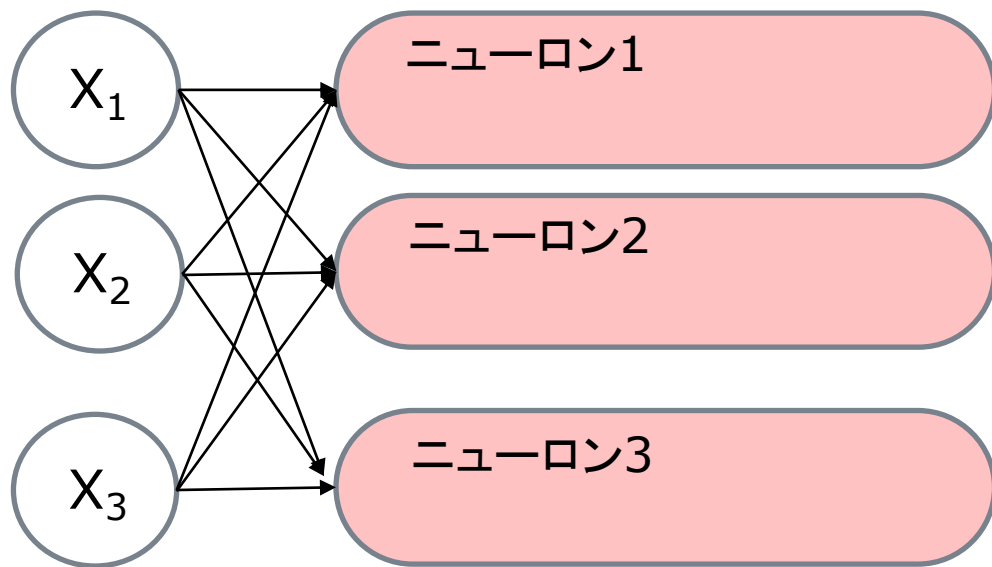
複数のニューロンが、一つの「層」を作っている



ここでは、3つのニューロンが一つの「層」を作っている。



入力 $X=[X_1, X_2, X_3]$



三つのニューロンで
その状態考えてみよう

例えば、入力 $X=[1,0,1]$ で

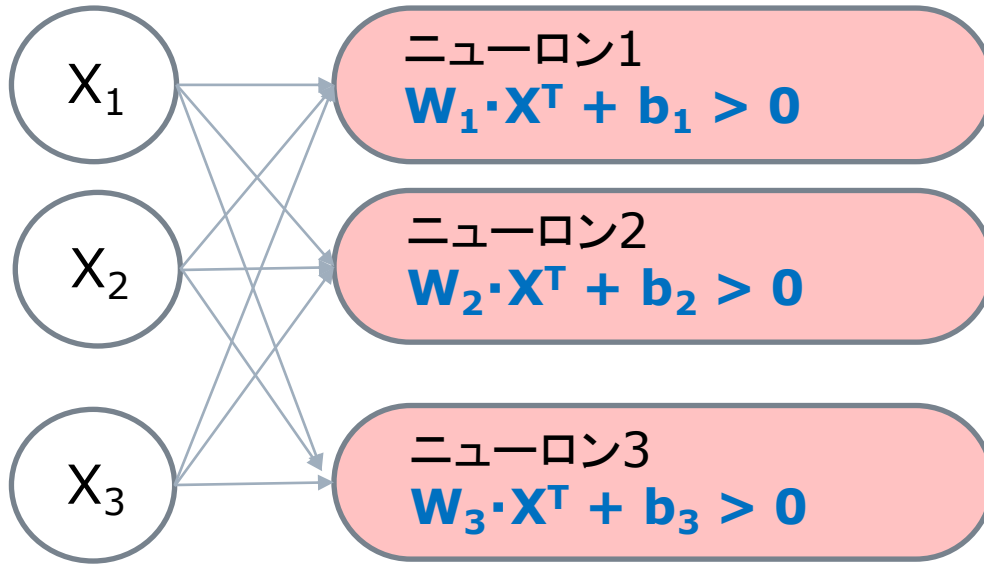
ニューロン1の重みが $[2,-3,4]$ バイアスが -4 なら
ニューロン1の状態は、 $2 \cdot 1 + (-3) \cdot 0 + 4 \cdot 1 - 4 = 2 > 0$ で発火。

ニューロン2の重みが $[-4,1,-5]$ バイアスが 5 なら
ニューロン2の状態は、 $(-4) \cdot 1 + 1 \cdot 0 + (-5) \cdot 1 + 5 = -4 < 0$ で発火せず。

ニューロン3の重みが $[-4,-2,5]$ バイアスが 2 なら
ニューロン3の状態は、 $(-4) \cdot 1 + (-2) \cdot 0 + 5 \cdot 1 + 2 = 3 > 0$ で発火。



入力 $X=[X_1, X_2, X_3]$



重み $\mathbf{W}_1=[W_{11}, W_{12}, W_{13}]$
バイアス b_1

重み $\mathbf{W}_2=[W_{21}, W_{22}, W_{23}]$
バイアス b_2

重み $\mathbf{W}_3=[W_{31}, W_{32}, W_{33}]$
バイアス b_3

一般的には、

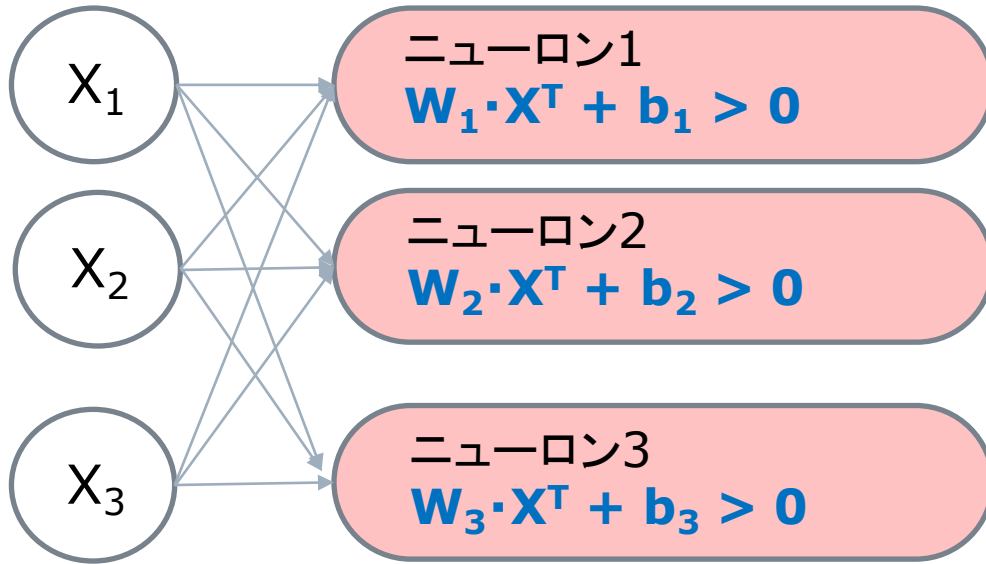
ニューロン1の状態: $\mathbf{W}_1 \cdot \mathbf{X}^T + b_1 = W_{11}X_1 + W_{12}X_2 + W_{13}X_3 + b_1$

ニューロン2の状態: $\mathbf{W}_2 \cdot \mathbf{X}^T + b_2 = W_{21}X_1 + W_{22}X_2 + W_{23}X_3 + b_2$

ニューロン3の状態: $\mathbf{W}_3 \cdot \mathbf{X}^T + b_3 = W_{31}X_1 + W_{32}X_2 + W_{33}X_3 + b_3$



入力 $X=[X_1, X_2, X_3]$



重み $W_1=[W_{11}, W_{12}, W_{13}]$
バイアス b_1

重み $W_2=[W_{21}, W_{22}, W_{23}]$
バイアス b_2

重み $W_3=[W_{31}, W_{32}, W_{33}]$
バイアス b_3

3つのニューロンの状態を並べた列ベクトル Y は、次のような値を持つ。

$$Y = \begin{bmatrix} W_1 \cdot X^T + b_1 \\ W_2 \cdot X^T + b_2 \\ W_3 \cdot X^T + b_3 \end{bmatrix} = \begin{bmatrix} W_{11}X_1 + W_{12}X_2 + W_{13}X_3 + b_1 \\ W_{21}X_1 + W_{22}X_2 + W_{23}X_3 + b_2 \\ W_{31}X_1 + W_{32}X_2 + W_{33}X_3 + b_3 \end{bmatrix}$$



ここで、重みの行ベクトルの要素を並べて、つぎのような行列 W を作る。

$$W = \begin{bmatrix} W_{11}, W_{12}, W_{13} \\ W_{21}, W_{22}, W_{23} \\ W_{31}, W_{32}, W_{33} \end{bmatrix}$$

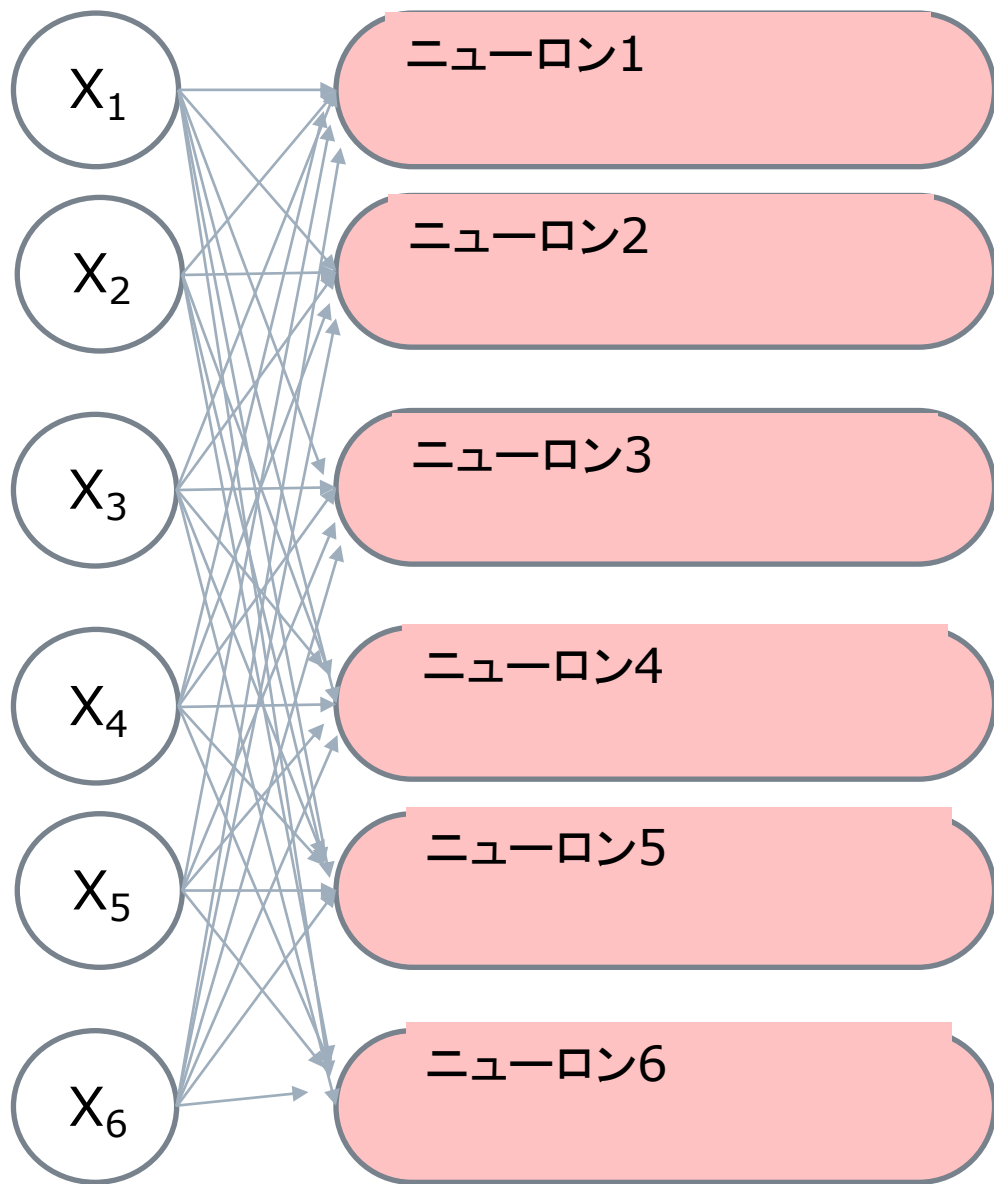
この時、次の式が成り立つことがわかる

$$\begin{bmatrix} W_{11}, W_{12}, W_{13} \\ W_{21}, W_{22}, W_{23} \\ W_{31}, W_{32}, W_{33} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} W_{11}X_1 + W_{12}X_2 + W_{13}X_3 + b_1 \\ W_{21}X_1 + W_{22}X_2 + W_{23}X_3 + b_2 \\ W_{31}X_1 + W_{32}X_2 + W_{33}X_3 + b_3 \end{bmatrix} = Y$$

Y は、 $W \cdot X^T + b^T$ の形をしている！



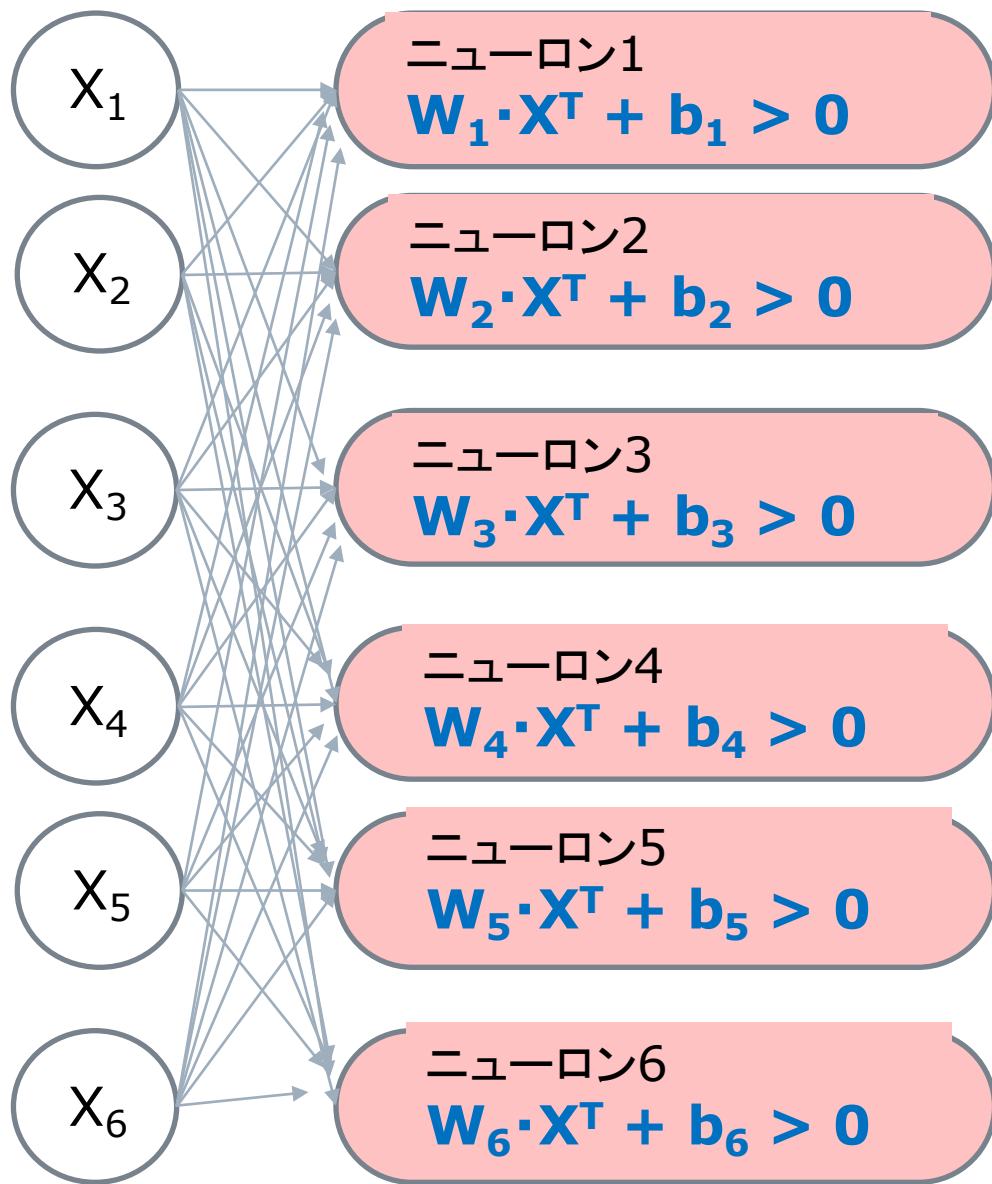
入力 $X=[X_1, X_2, X_3, X_4, X_5, X_6]$



6つのニューロンで
考えてみよう。
基本的には、同じで
ある。



入力 $X=[X_1, X_2, X_3, X_4, X_5, X_6]$



重み $W_1=[W_{11}, W_{12}, W_{13}, W_{14}, W_{15}, W_{16}]$
バイアス b_1

重み $W_2=[W_{21}, W_{22}, W_{23}, W_{24}, W_{25}, W_{26}]$
バイアス b_2

重み $W_3=[W_{31}, W_{32}, W_{33}, W_{34}, W_{35}, W_{36}]$
バイアス b_3

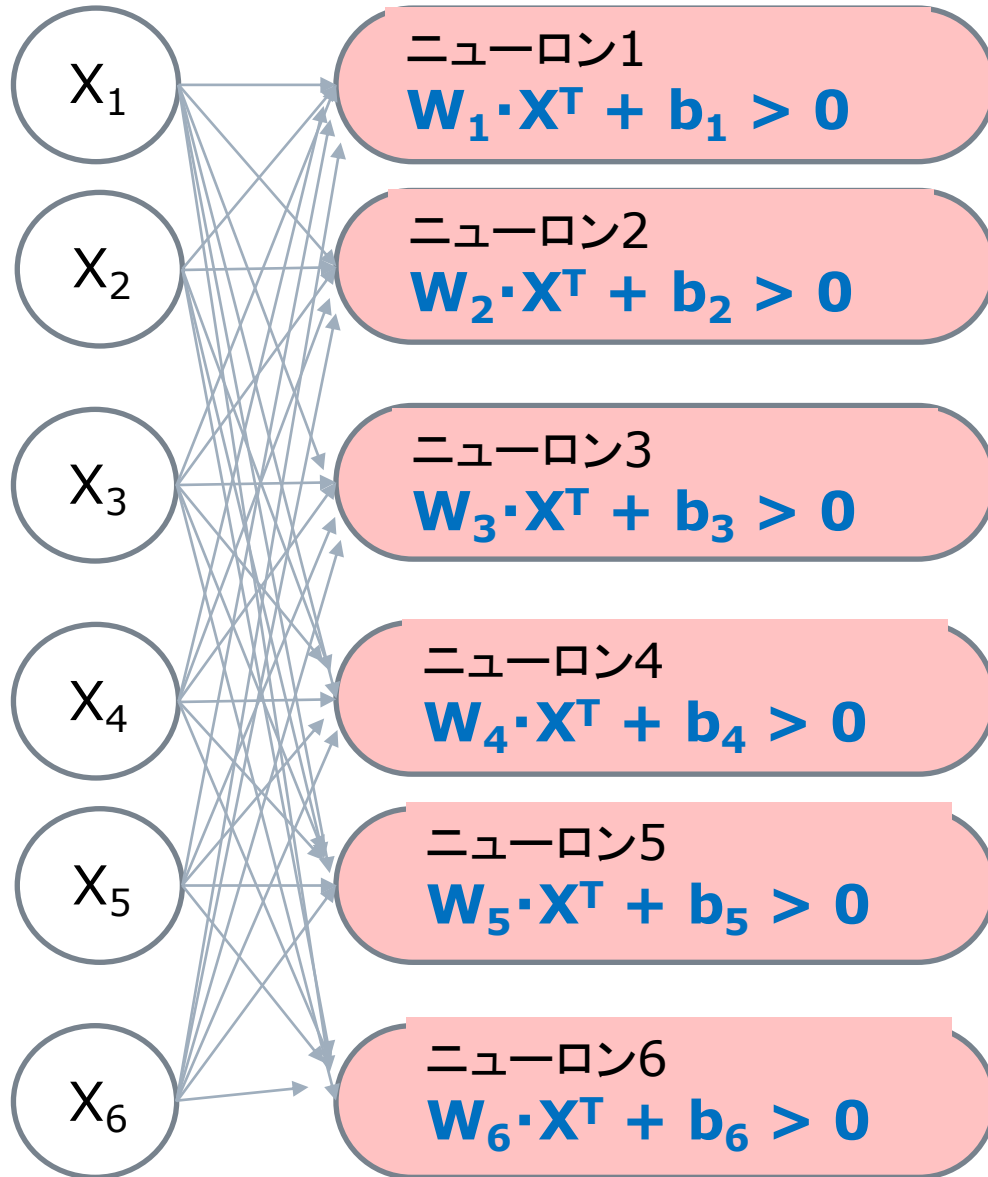
重み $W_4=[W_{41}, W_{42}, W_{43}, W_{44}, W_{45}, W_{46}]$
バイアス b_4

重み $W_5=[W_{51}, W_{52}, W_{53}, W_{54}, W_{55}, W_{56}]$
バイアス b_5

重み $W_6=[W_{61}, W_{62}, W_{63}, W_{64}, W_{65}, W_{66}]$
バイアス b_6



入力 $X=[X_1, X_2, X_3, X_4, X_5, X_6]$



値: $W_{11}X_1 + W_{12}X_2 + W_{13}X_3 +$
 $W_{14}X_4 + W_{15}X_5 + W_{16}X_6 + b_1$

値: $W_{21}X_1 + W_{22}X_2 + W_{23}X_3 +$
 $W_{24}X_4 + W_{25}X_5 + W_{26}X_6 + b_2$

値: $W_{31}X_1 + W_{32}X_2 + W_{33}X_3 +$
 $W_{34}X_4 + W_{35}X_5 + W_{36}X_6 + b_3$

値: $W_{41}X_1 + W_{42}X_2 + W_{43}X_3 +$
 $W_{44}X_4 + W_{45}X_5 + W_{46}X_6 + b_4$

値: $W_{51}X_1 + W_{52}X_2 + W_{53}X_3 +$
 $W_{54}X_4 + W_{55}X_5 + W_{56}X_6 + b_5$

値: $W_{61}X_1 + W_{62}X_2 + W_{63}X_3 +$
 $W_{64}X_4 + W_{65}X_5 + W_{66}X_6 + b_6$



行列による表現(六個のニューロンの場合)

$$\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} & W_{16} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} & W_{26} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} & W_{36} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} & W_{46} \\ W_{51} & W_{52} & W_{53} & W_{54} & W_{55} & W_{56} \\ W_{61} & W_{62} & W_{63} & W_{64} & W_{65} & W_{66} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} =$$

$$\begin{bmatrix} W_{11}X_1 + W_{12}X_2 + W_{13}X_3 + W_{14}X_4 + W_{15}X_5 + W_{16}X_6 + b_1 \\ W_{21}X_1 + W_{22}X_2 + W_{23}X_3 + W_{24}X_4 + W_{25}X_5 + W_{26}X_6 + b_2 \\ W_{31}X_1 + W_{32}X_2 + W_{33}X_3 + W_{34}X_4 + W_{35}X_5 + W_{36}X_6 + b_3 \\ W_{41}X_1 + W_{42}X_2 + W_{43}X_3 + W_{44}X_4 + W_{45}X_5 + W_{46}X_6 + b_4 \\ W_{51}X_1 + W_{52}X_2 + W_{53}X_3 + W_{54}X_4 + W_{55}X_5 + W_{56}X_6 + b_5 \\ W_{61}X_1 + W_{62}X_2 + W_{63}X_3 + W_{64}X_4 + W_{65}X_5 + W_{66}X_6 + b_6 \end{bmatrix}$$

これも、 $\mathbf{W} \cdot \mathbf{X}^T + \mathbf{b}^T$ の形をしている！



式は便利だ！

$$Y = W \cdot X^T + b^T \text{ が表すもの}$$

数学的な表記を使うと、問題を簡潔に定式化できる。
ここでは、一つのニューロンの状態だけでなく、複数のニューロンからなる層も、一つの式で表現できることについて述べる。



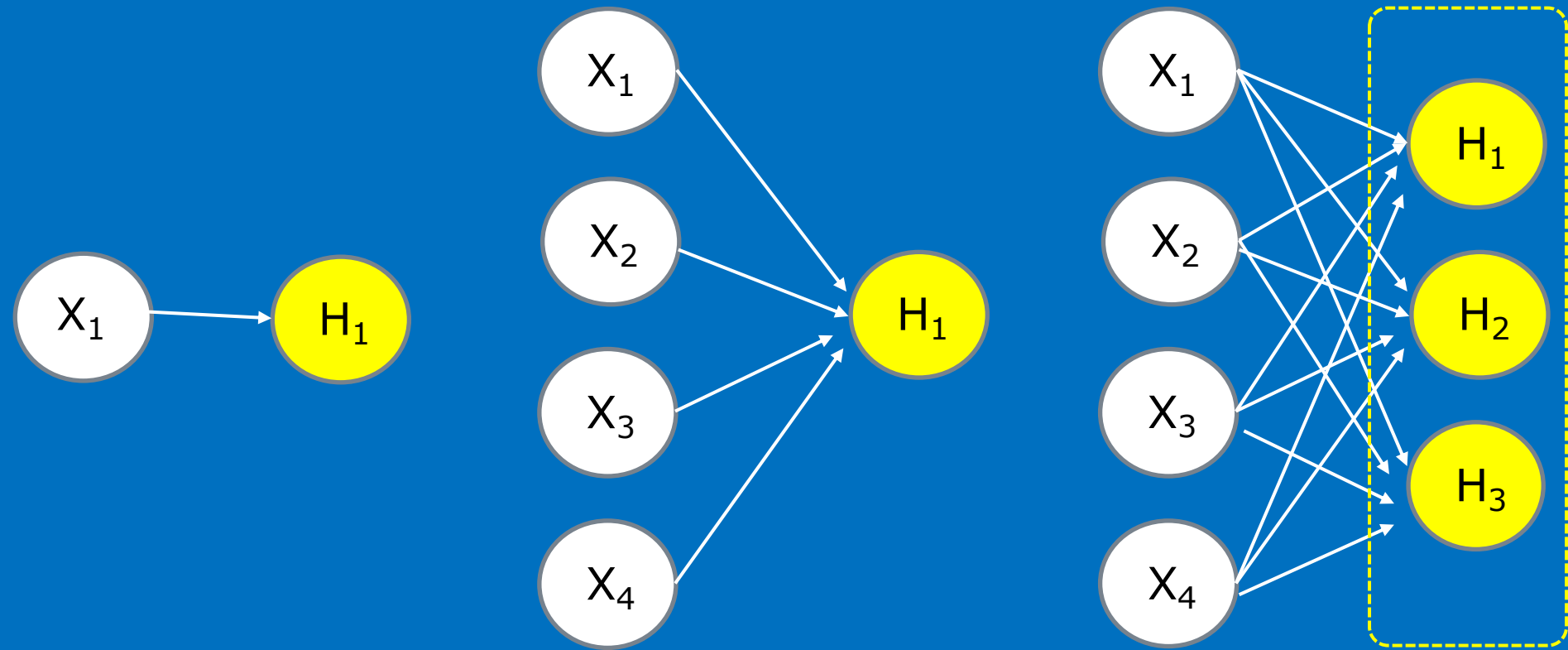
式は便利だ！

$$Y = W \cdot X^T + b^T \text{ が表すもの}$$

- ニューラル・ネットワークの本を読んで、最初につまずくのは、行列が出てくるあたりだと思う。ただ、ニューラル・ネットワークの理解で、当面、必要なことは、式 $Y = W \cdot X^T + b$ が表現しているものを理解することだ。見慣れていないだけで、難しい数学ではない。
- 具体的な例で、式 $Y = W \cdot X^T + b^T$ が表しているものを、紹介したいと思う。一見、複雑に見える沢山の関係が、この式一つで表現されていることがわかるはずだ。式は便利だということがわかってもらえると思う。
- 以下の例では、 X も Y も b も、最初から行ベクトルだとしよう（このことを、仮に、 $X = [X_1, X_2, \dots, X_n]^T$ のように表そう）。この時、 $Y = W \cdot X^T + b^T$ が、何を表すかを考えよう。



$$W \cdot X^T + b^T$$

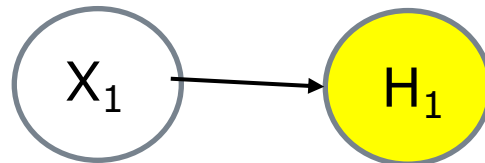


単純なものでも複雑なものでも
ネットワークの状態は同じ式で表現できる



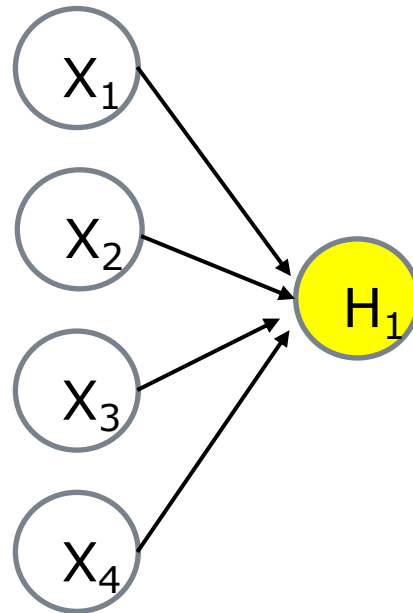
$Y = W \cdot X^T + b^T$ が表すもの W, X, b が数値 (スカラー) の時

- $Y=y, X=x, W=a, b=b$ なら、この式は、 $y = ax + b$ という、式を表す。
- $Y=y, X=x, W=2, b=3$ なら、この式は、 $y = 2x + 3$ という、式を表す。
- $Y=y, X=x, W=4, b=0$ なら、この式は、 $y = 4x$ という、式を表す。



$Y = W \cdot X^T + b^T$ が表すもの W, X がベクトルの時

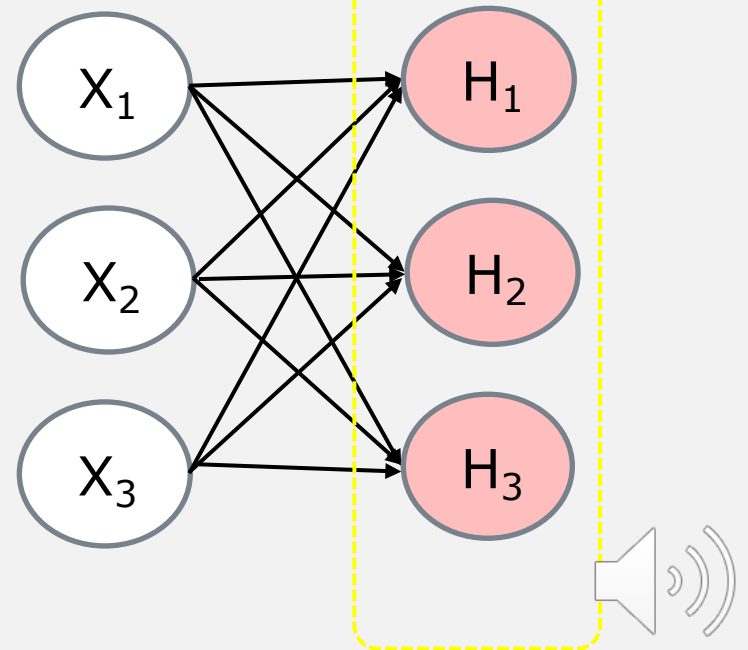
- $Y=y, X=[x_1, x_2]^T, W=[w_1, w_2], b=b_0$ なら、この式は、
 $y = w_1 \cdot x_1 + w_2 \cdot x_2 + b_0$ という、式を表す。
- $Y=y, X=[c, d]^T, W=[m, n], b=b$ なら、この式は、
 $y = m \cdot c + n \cdot d + b$ という、式を表す。



$Y = W \cdot X^T + b^T$ が表すもの Wが行列, X, bがベクトルの時

- 3x3 (3行3列の行列)の行列を、 $\begin{bmatrix} [d,e,f], [g,h,i], [j,k,l] \end{bmatrix}$ と表そう。
- $Y = [y_1, y_2, y_3]^T$, $X = [x_1, x_2, x_3]^T$, $W = \begin{bmatrix} [2,3,-1], [4,-5,1], [1,2,3] \end{bmatrix}$, $b = [1,2,3]^T$ の時、 $Y = W \cdot X^T + b^T$ は、次の三つの式を表す。

- $y_1 = 2x_1 + 3x_2 - x_3 + 1$
- $y_2 = 4x_1 + 5x_2 + x_3 + 2$
- $y_3 = x_1 + 2x_2 + 3x_3 + 3$



$Y = W \cdot X^T + b^T$ が表すもの

- 例えば、 W が 6×6 (6行6列)の行列の時、 $Y = W \cdot X^T + b^T$ が表すものの例については、先に見た。もちろん n が100でも10,000でも同じ式で、表現できる。
- W が n 行 m 列の行列の時、 $Y = W \cdot X^T + b^T$ が、 n 個の式を表していることを示すために、次のように書くこともある。 $Y_n = W_n \cdot X + b_n$
- こうした表記は、数百・数千のノードからなるニューラル・ネットワークが満たす多数の関係式を、一つの式で簡潔にまとめることができるので、プログラミング上も極めて有用である。



ニューロンが発火する条件

$$W \cdot X^T + b^T > 0$$

- **一つのニューロンで、一つの入力の場合。**
一つの入力を X (実数)、そのシナプスの重みを W (実数)、ニューロンのバイアスを b (実数)とすると、条件は

$$W \cdot X + b > 0$$

$W \cdot X$ は、普通の掛け算

- **一つのニューロンで、複数の入力の場合。**

複数の入力を \mathbf{X} (ベクトル)、その複数のシナプスの重みを \mathbf{W} (ベクトル)、ニューロンのバイアスを b (実数)とすると、条件は

$$\mathbf{W} \cdot \mathbf{X}^T + b > 0$$

$\mathbf{W} \cdot \mathbf{X}^T$ は、ベクトルの内積

- **複数のニューロンの一つの層で、複数の入力の場合**

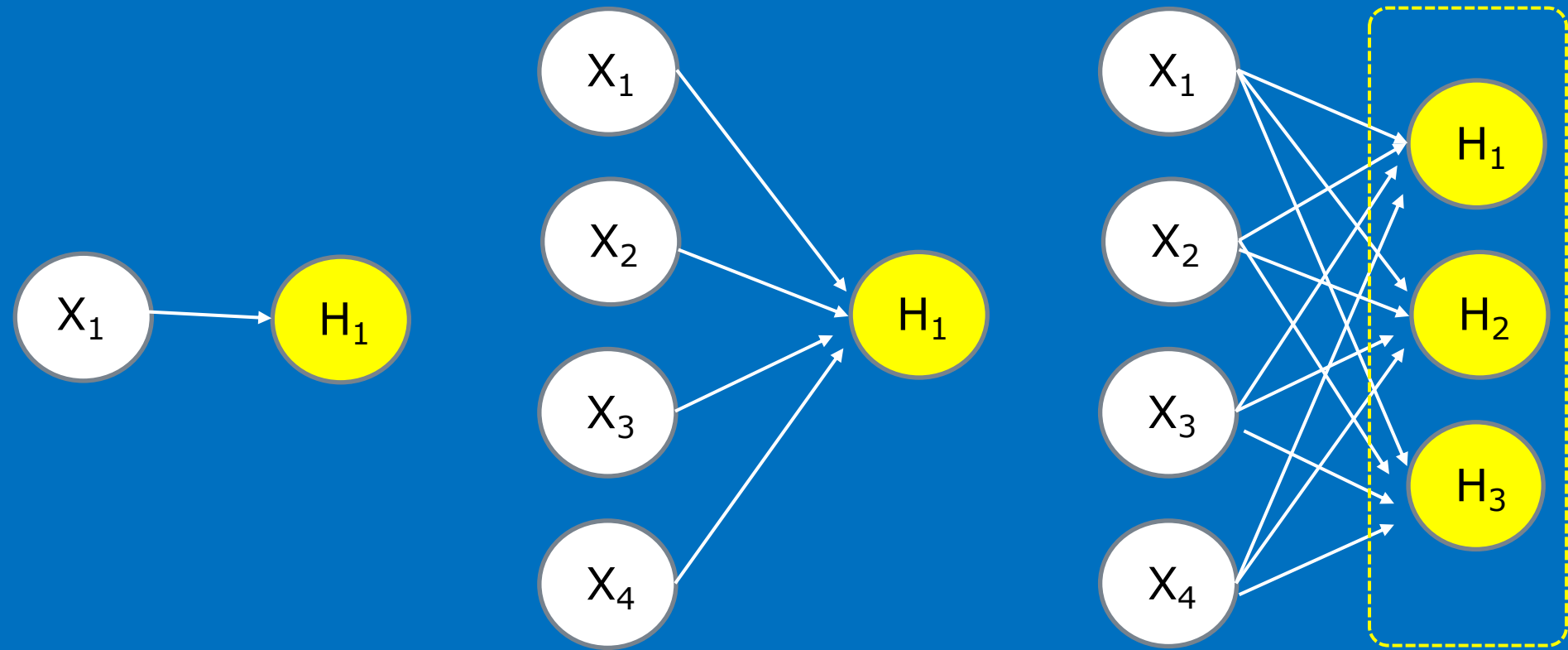
複数の入力を \mathbf{X} (ベクトル)、その層内のシナプスの重み全体を \mathbf{W} (行列)、ニューロンの複数のバイアスを \mathbf{b} (ベクトル)とすると、条件は

$$\mathbf{W} \cdot \mathbf{X}^T + \mathbf{b}^T > 0$$

$\mathbf{W} \cdot \mathbf{X}^T$ は、行列とベクトルの積



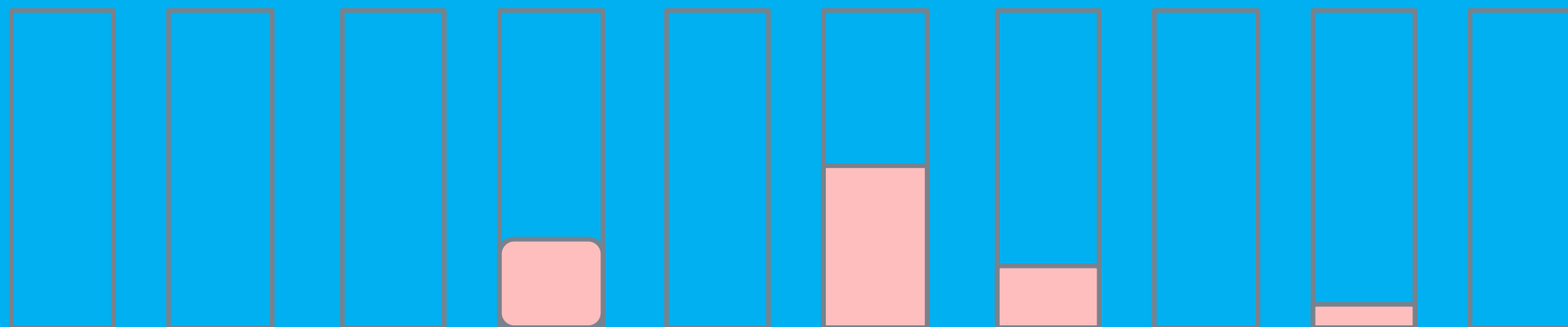
$$W \cdot X^T + b^T$$



単純なものでも複雑なものでも
ネットワークの状態は同じ式で表現できる



softmax 関数の出力例



ActivatorとSoftMax

ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9

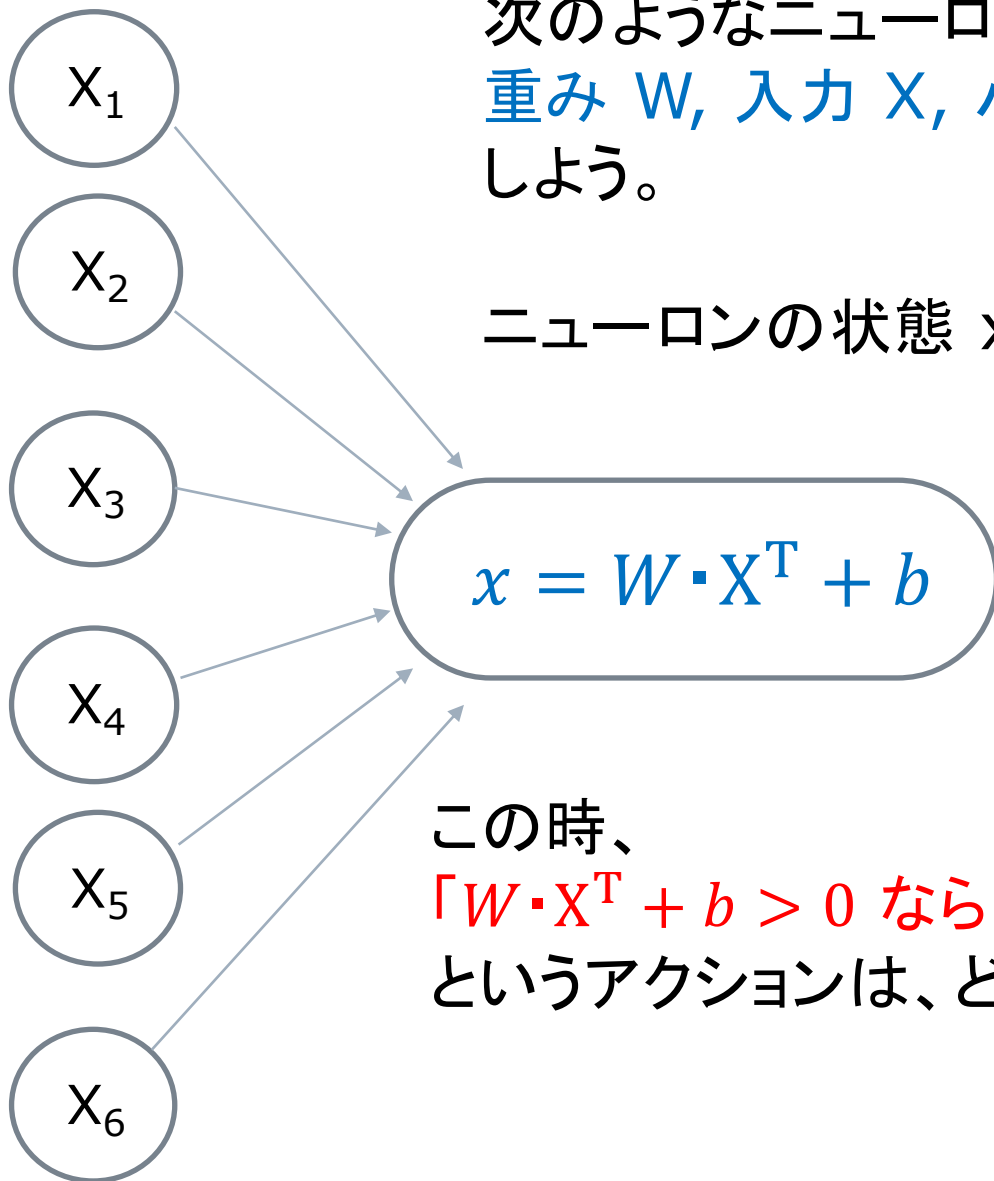


ニューロンを動かす

-- 活性化関数 (Activator)

活性化関数は、その名前の通りに、一つのニューロンが発火(活性化)する時の条件と出力を関数の形で表したものである。いろいろな種類の活性化関数がある。





次のようなニューロンのネットワークと、その重み W , 入力 X , バイアス b が与えられたとしよう。

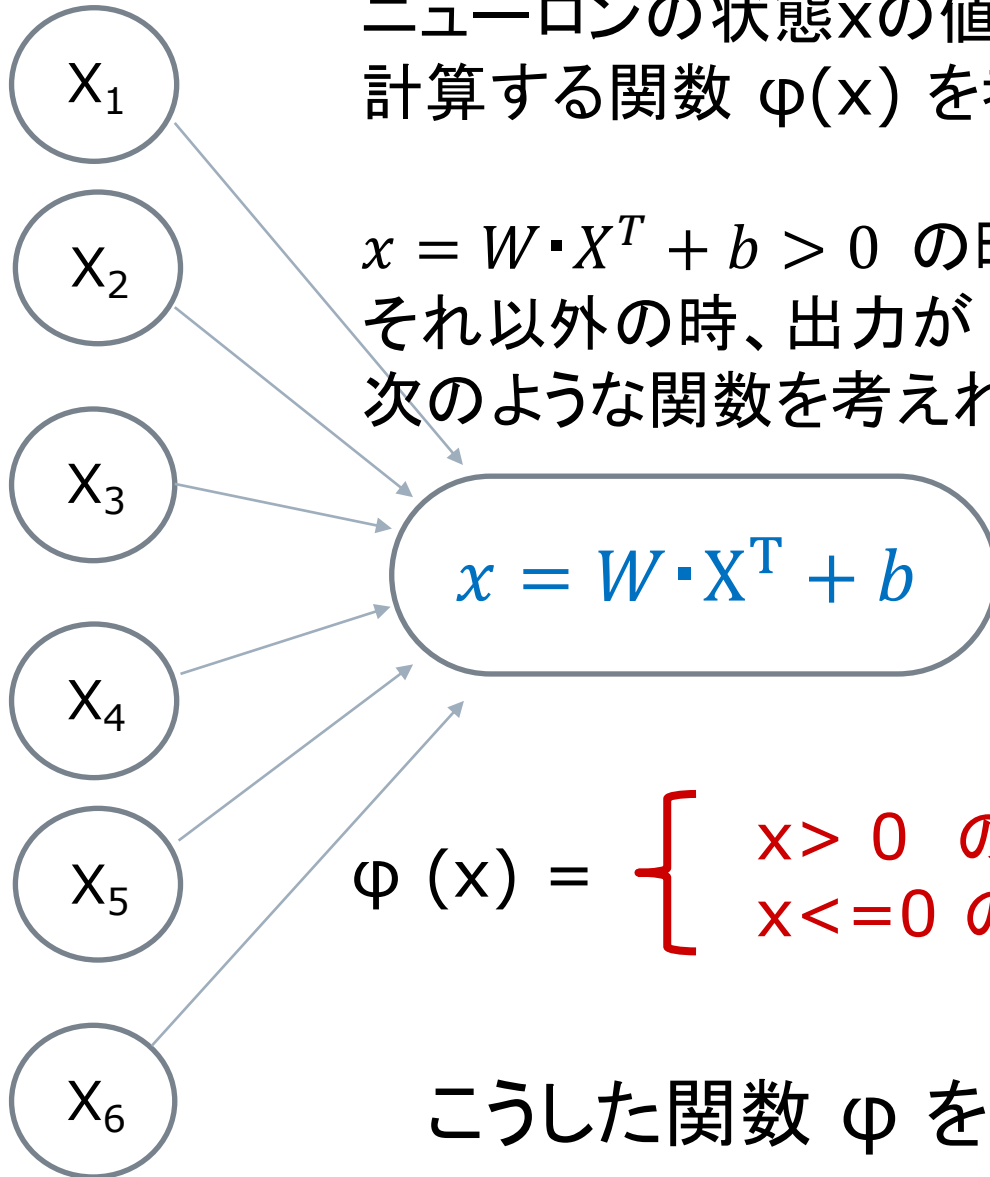
ニューロンの状態 x は、次の式で計算される。

この時、
「 $W \cdot X^T + b > 0$ ならこのニューロンは発火する」
というアクションは、どのように表現できるだろうか？



ニューロンの状態 x の値に応じてニューロンの出力を計算する関数 $\varphi(x)$ を考える。

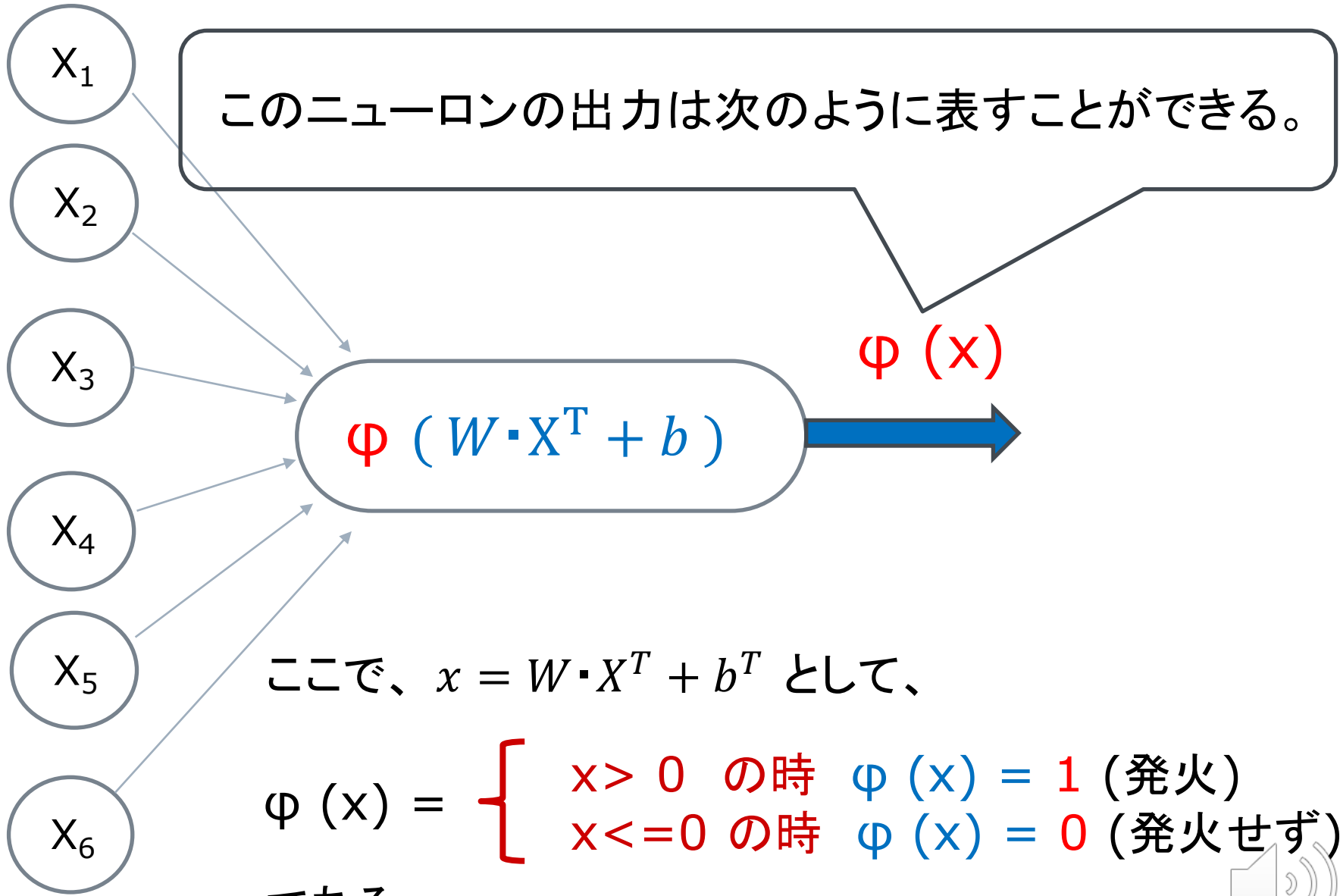
$x = W \cdot X^T + b > 0$ の時、出力が 1(発火)、
それ以外の時、出力が 0(発火せず)だとすれば、
次のような関数を考えればいい。



$$\varphi(x) = \begin{cases} x > 0 \text{ の時} & \varphi(x) = 1 \text{ (発火)} \\ x \leq 0 \text{ の時} & \varphi(x) = 0 \text{ (発火せず)} \end{cases}$$

こうした関数 φ を、「**活性化関数**」とよぶ。





ここで、 $x = W \cdot X^T + b^T$ として、

$$\varphi(x) = \begin{cases} x > 0 \text{ の時} & \varphi(x) = 1 \text{ (発火)} \\ x \leq 0 \text{ の時} & \varphi(x) = 0 \text{ (発火せず)} \end{cases}$$

である。



入力 X , 重み W , バイアス b の
ニューロンの出力は、
Activator ϕ をつかって、

$\phi(W \cdot X^T + b)$ の形で表せる。



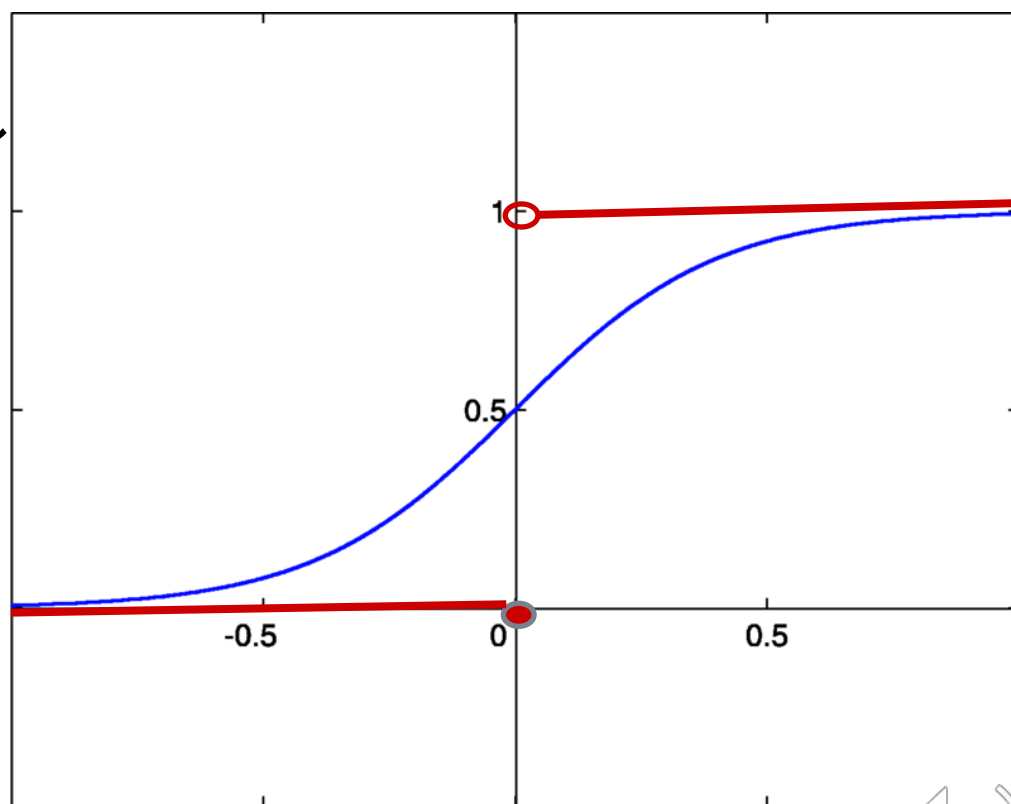
さまざまなタイプのActivator

先に紹介したActivator $\varphi(x)$ は、 $x=0$ で値が非連続に変化する。それは連続でも微分可能でもない。実際の応用では、それらに近似した連続で微分可能な関数をActivatorとしてを利用することがある。また、それが返す値も、0と1だけとは限らない。



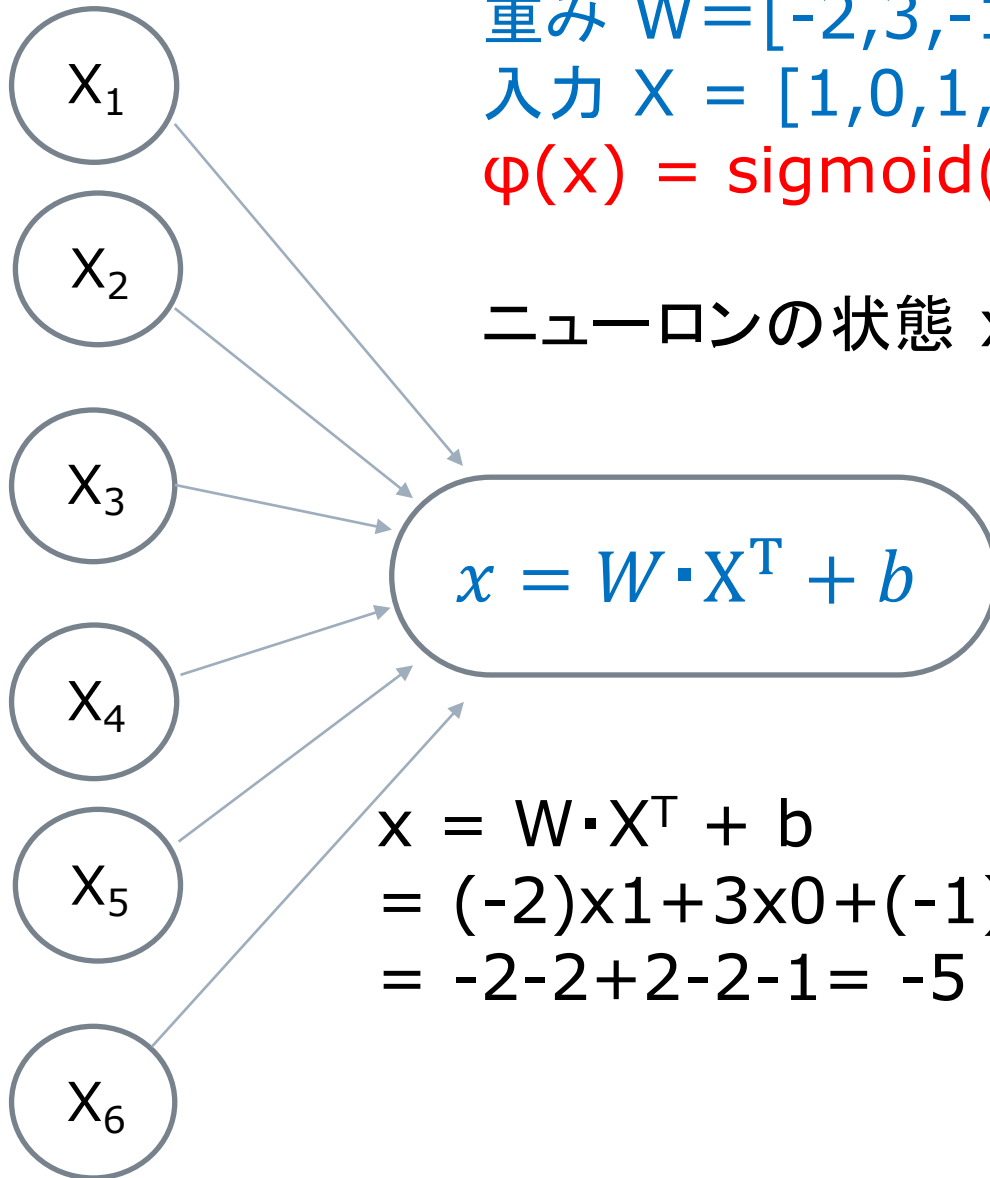
sigmoid 関数 (logistic 関数)

- 先の φ を、右のグラフでは、赤線で示した。 $x=0$ の時に、非連続に変化する。
- 実際の応用では、こうした飛躍のない、青線のような関数が持たれることが多い。
- これをsigmoid関数という。(logistic関数)
- 近似的には、
 $x < 0$ の時 0
 $x > 0$ の時 1
基本的には0と1の間の値をとる。



重み $W = [-2, 3, -1, 3, 2, -2]$ 、バイアス $b = -1$
入力 $X = [1, 0, 1, 0, 1, 1]$ として、Activatorに
 $\varphi(x) = \text{sigmoid}(x)$ を使ってみよう。

ニューロンの状態 x は、次の式で計算される。

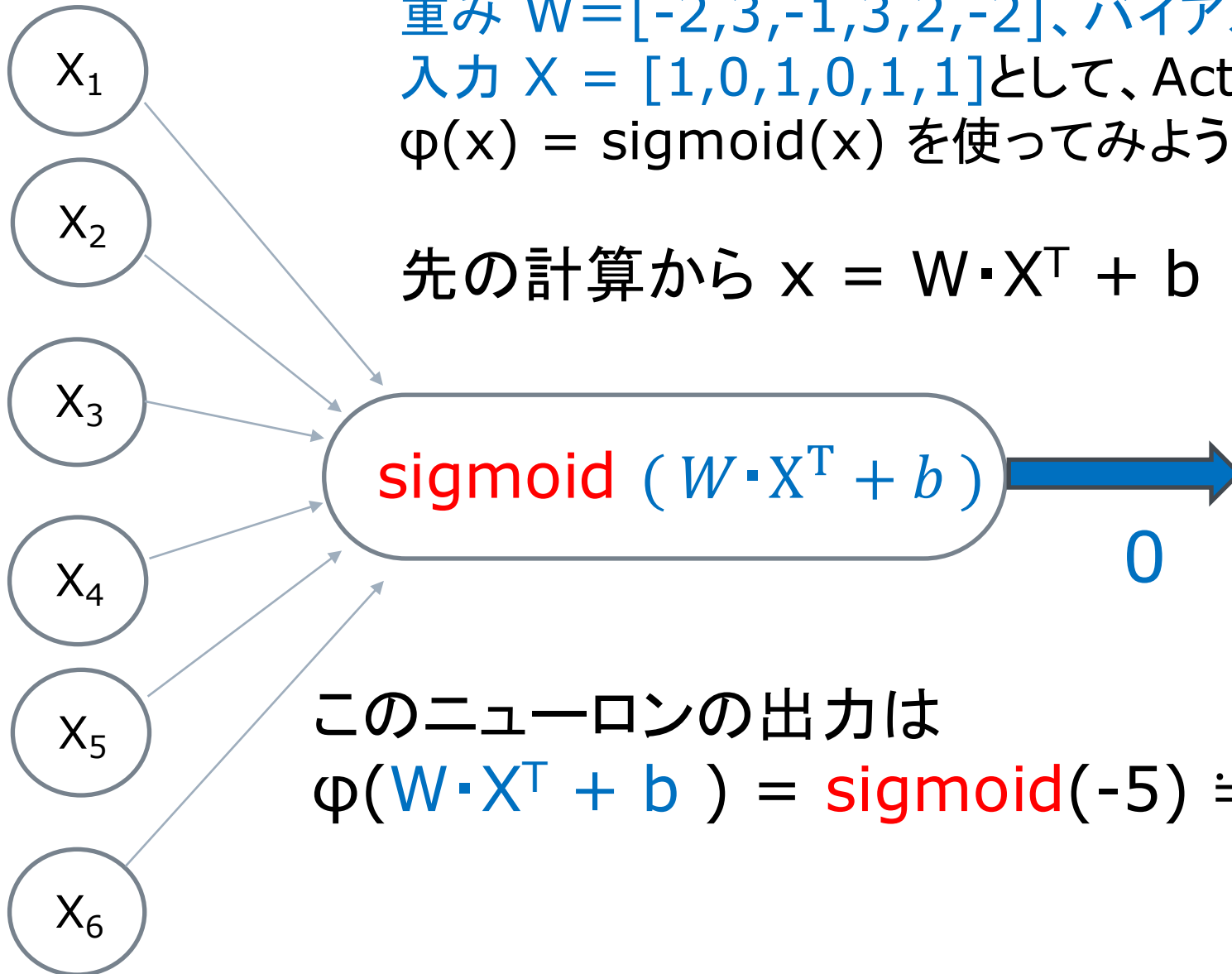


$$\begin{aligned} x &= W \cdot X^T + b \\ &= (-2) \times 1 + 3 \times 0 + (-1) \times 1 + 3 \times 0 + 2 \times 1 + (-2) \times 1 - 1 \\ &= -2 - 2 + 2 - 2 - 1 = -5 \end{aligned}$$



重み $W = [-2, 3, -1, 3, 2, -2]$ 、バイアス $b = -1$
入力 $X = [1, 0, 1, 0, 1, 1]$ として、Activatorに
 $\varphi(x) = \text{sigmoid}(x)$ を使ってみよう。

先の計算から $x = W \cdot X^T + b = -5$

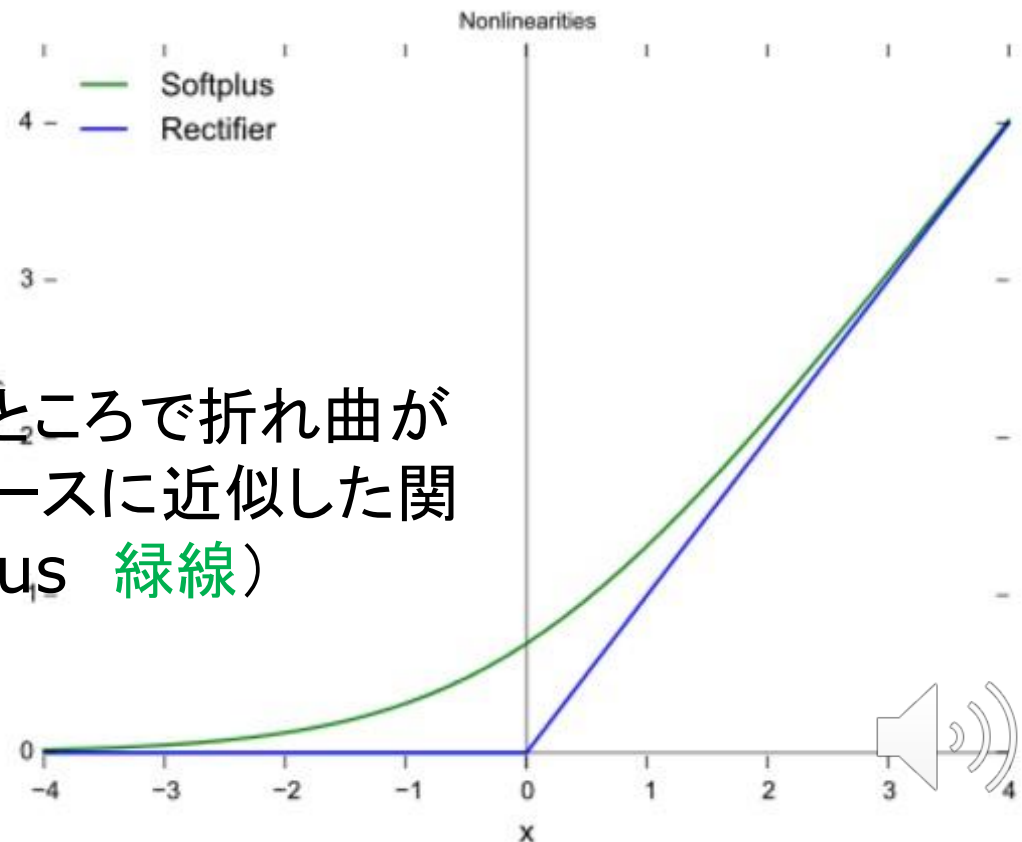


このニューロンの出力は
 $\varphi(W \cdot X^T + b) = \text{sigmoid}(-5) \doteq 0$



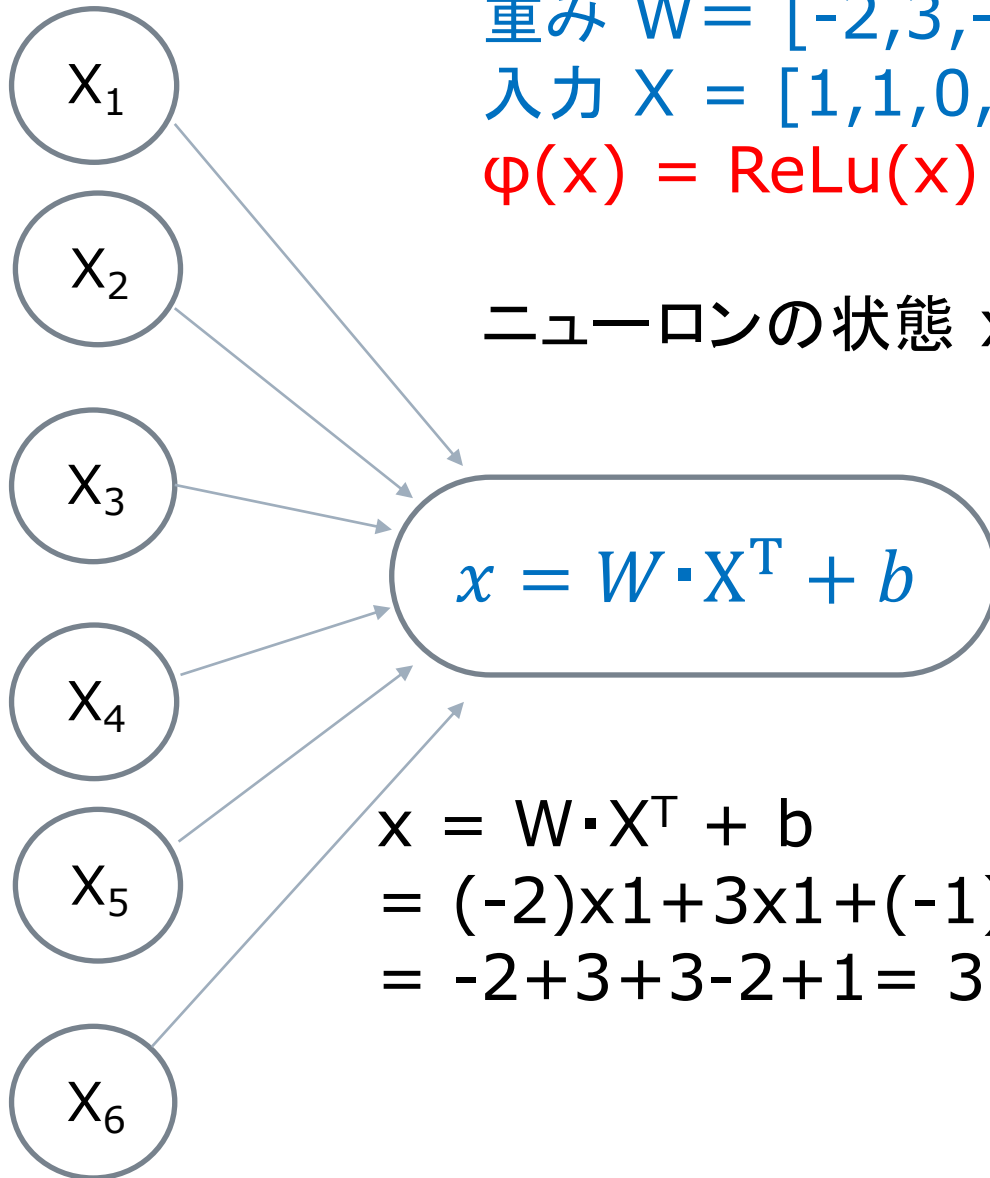
ReLU (rectified linear unit)

- rectifier は、「整流器」。ReLUは、マイナスの値は通さず0を返すが、プラスの値は、そのまま通す。(青線)
- sigmoidは、0と1の間の値を出力するが、ReLUの出力には、そうした制限はない。
- 断続はないが、 $x=0$ のところで折れ曲がっている。それを、スムーズに近似した関数も存在する。(Softplus 緑線)
- $x < 0$ の時 0
 $x > 0$ の時 x
の値をとる。



重み $W = [-2, 3, -1, 3, 2, -2]$ 、バイアス $b = -1$
入力 $X = [1, 1, 0, 1, 0, 1]$ として、Activatorに
 $\varphi(x) = \text{ReLu}(x)$ を使ってみよう。

ニューロンの状態 x は、次の式で計算される。

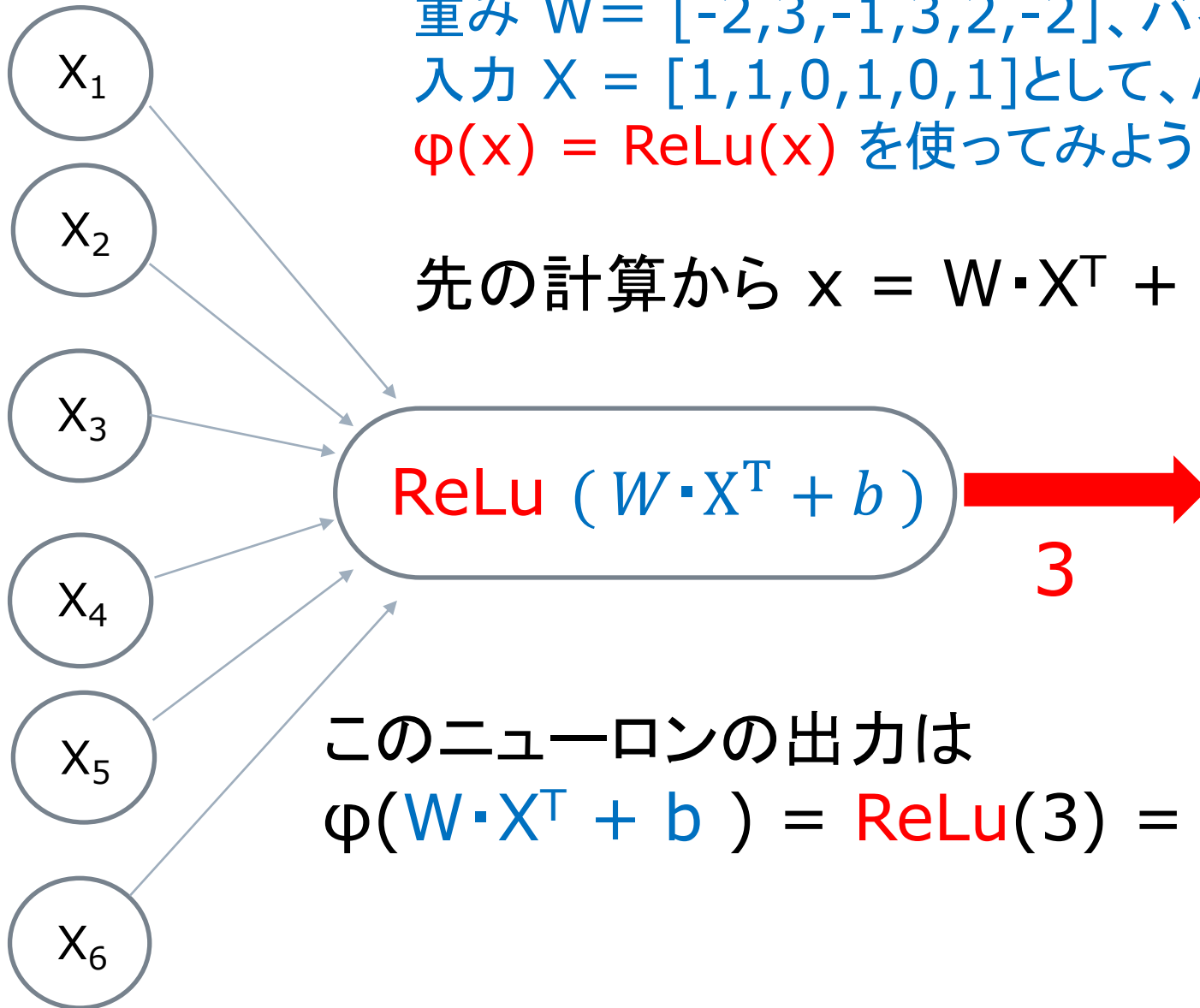


$$\begin{aligned}x &= W \cdot X^T + b \\ &= (-2) \times 1 + 3 \times 1 + (-1) \times 0 + 3 \times 1 + 2 \times 0 + (-2) \times 1 + 1 \\ &= -2 + 3 + 3 - 2 + 1 = 3\end{aligned}$$



重み $W = [-2, 3, -1, 3, 2, -2]$ 、バイアス $b = -1$
入力 $X = [1, 1, 0, 1, 0, 1]$ として、Activatorに
 $\varphi(x) = \text{ReLu}(x)$ を使ってみよう。

先の計算から $x = W \cdot X^T + b = 3$

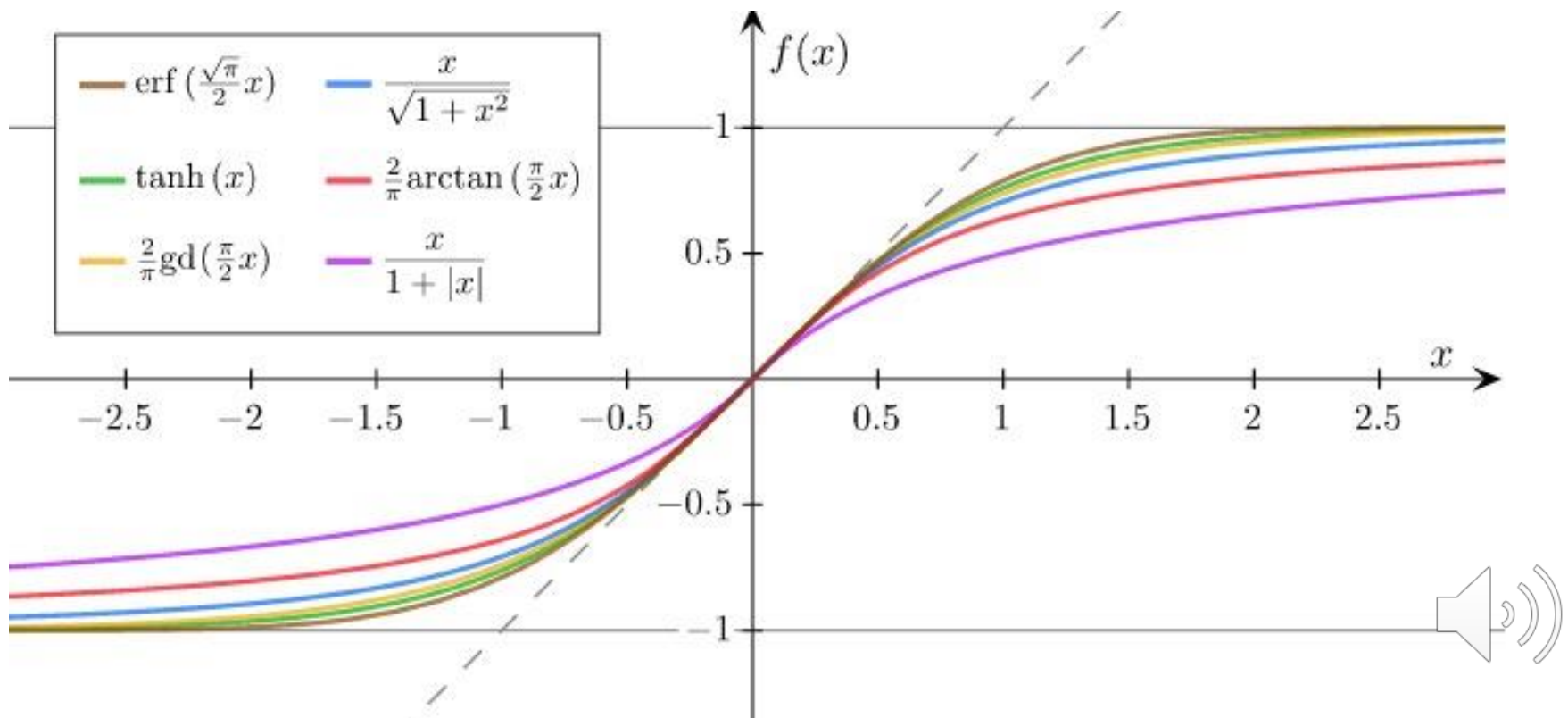


このニューロンの出力は
 $\varphi(W \cdot X^T + b) = \text{ReLu}(3) = 3$



tanh 関数

- sigmoidの仲間だが、ちょうどそれを1/2ほど、下に移動した形である。**-1と+1との間の値を出力する**。tanh以外にも、同じ性質を持つ、様々な関数が提案されている。



クラス分けのActivator SoftMax

画像認識等のクラス分けの出力には、Softmaxという活性化関数がよく利用される



softmax 関数

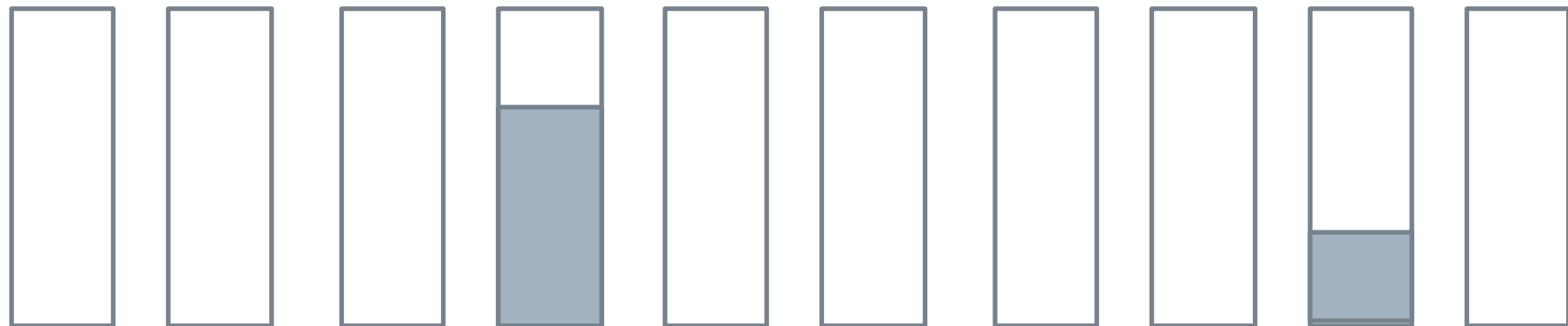
- クラス分けの場合の出力に利用されるactivator。

手書き数字画像の認識だと、提示された画像が0から9までの10個の数字のどれかだと判断しなければいけない。

- ニューラル・ネットワークでは、こうした場合、出力層に、0から9までの10個の数字に対応した10個のノードが並び、それぞれのノードの値が、「提示された画像がその数字である確率」であるようにする。
- ノードごとに、確率を計算するのが softmax関数である。最終的には、10個のノードの中で、その確率が最大であるものが選ばれる。



softmax 関数の出力例



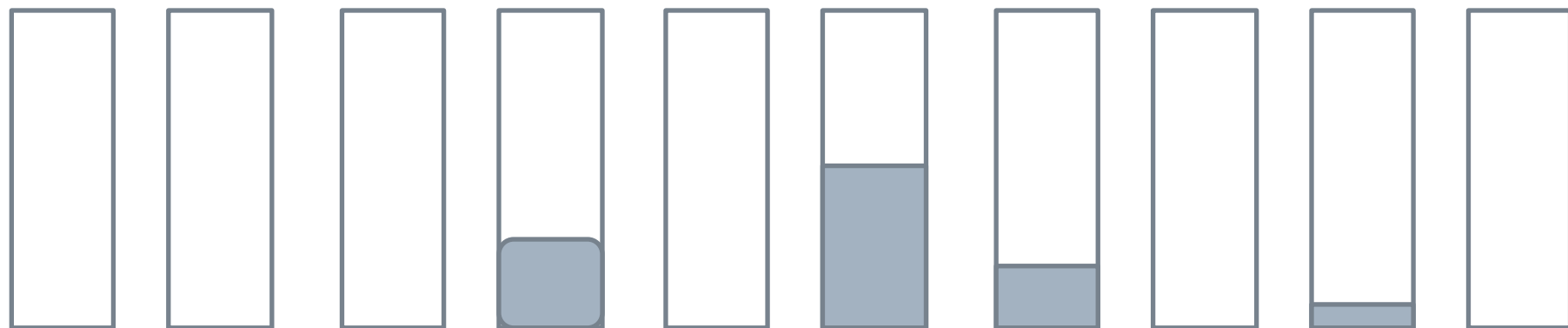
ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9





softmax 関数の出力例

0 0 0 0 0 **1** 0 0 0 0

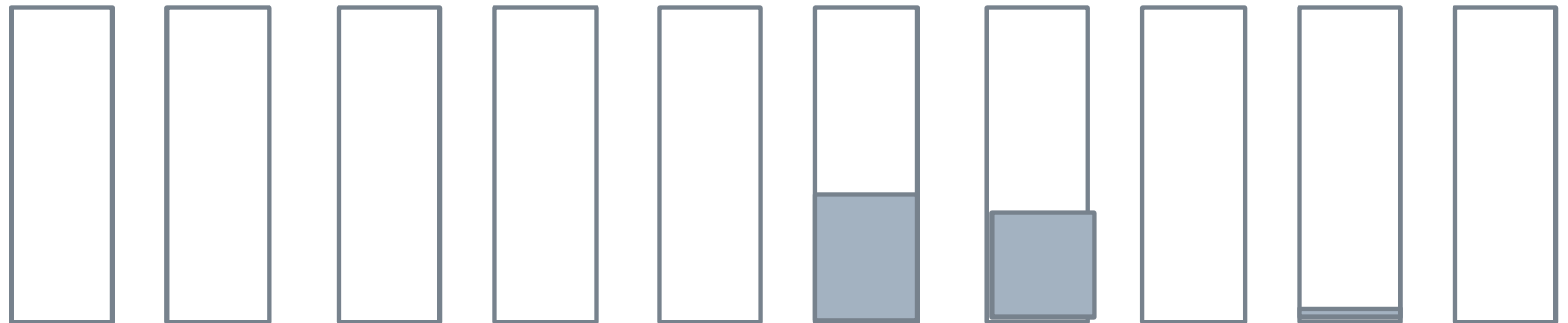


ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9





softmax 関数の出力例



ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9



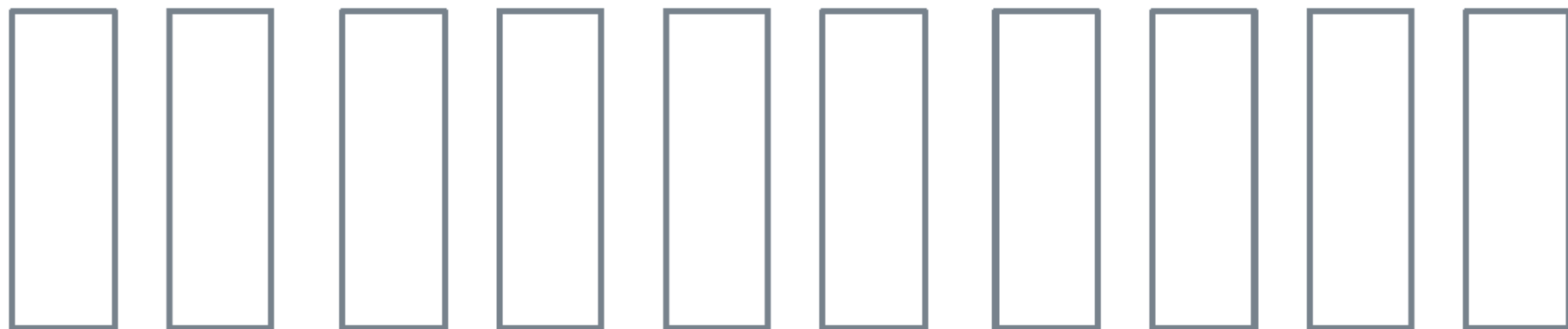
softmax 関数

- これまで見たActivatorは、一つのニューロンについて、その状態をそのニューロンの出力に変えるものだったが、softmaxは少し違っている。それは、複数のニューロンの状態にまたがって作用するActivatorで、しかも、複数のニューロンごとに出力を行う。
- あと補足が必要なのだが、先の図は、softmax関数と、その出力から最大値を選ぶ、maxarg関数の二つの関数の働きが示されている。
- ここでは、softmaxが作用するノードの数を10個だとして、説明を始めたい。



10個のノードの状態の値を、
 x_0, x_1, \dots, x_9 とする

0 0 0 0 0 0 0 0 0 0



x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

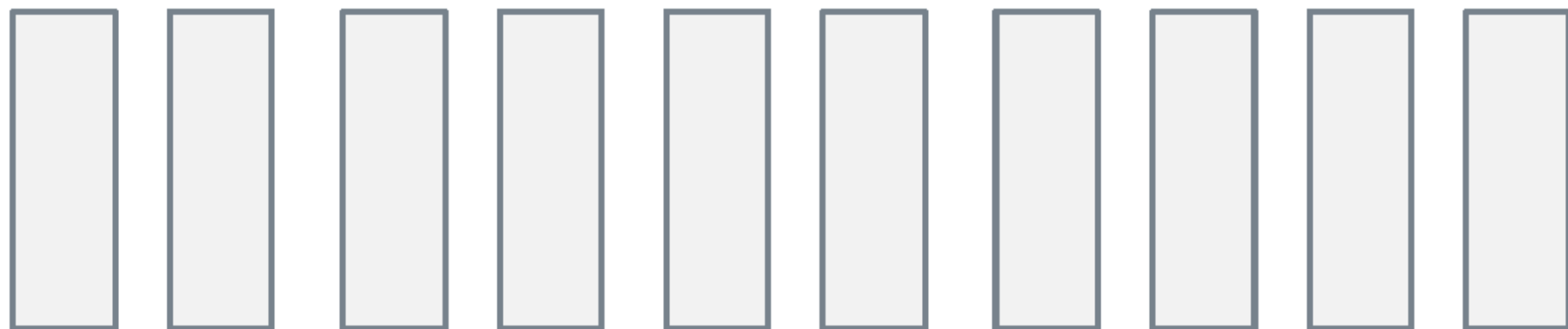
ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9



ベクトル $[x_0, x_1, \dots, x_9]$ に対して、
softmaxを適用する

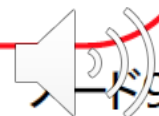
0 0 0 0 0 0 0 0 0 0

softmax($[x_0, x_1, \dots, x_9]$)



x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

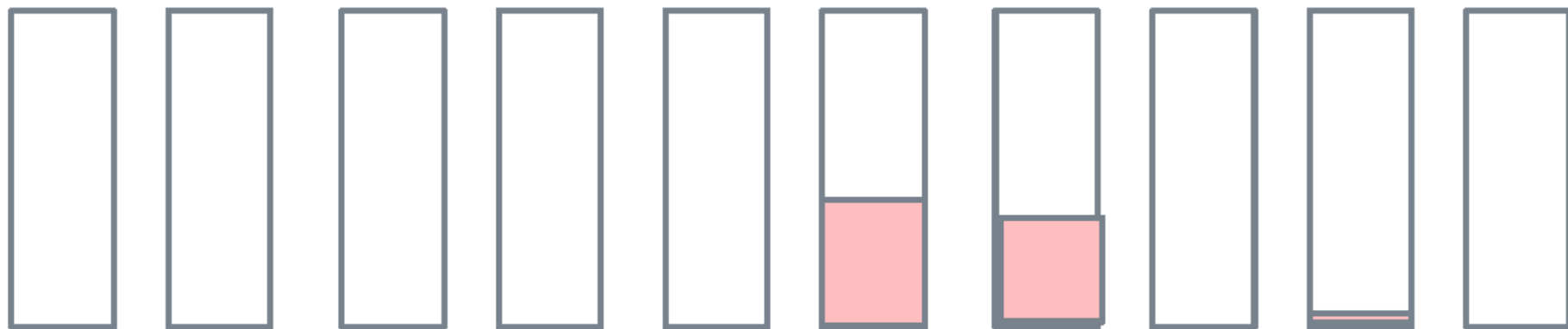
ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9



softmax($[x_0, x_1, \dots, x_9]$)の結果に対して
最大値をとる値を選ぶargmax関数を適用する

argmax(softmax($[x_0, x_1, \dots, x_9]$)))

0 0 0 0 0 1 0 0 0 0



x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

ノード0 ノード1 ノード2 ノード3 ノード4 ノード5 ノード6 ノード7 ノード8 ノード9



softmax関数の定義

- N次元のベクトル $[x_0, x_1, \dots, x_N]$ に対して、 $\text{softmax}([x_0, x_1, \dots, x_N])$ は、N次元のベクトルを返す。
- $\text{softmax}([x_0, x_1, \dots, x_N])$ が返すN次元のベクトルの*i*番目の要素をと s_i すると、

$$s_i = \frac{e^{x_i}}{\sum_{j=0}^{N-1} e^{x_j}}$$

である。

- 分母のブルーの部分は、全ての計算で共通である。



softmax関数の pythonでの計算

- 7次元のベクトル

a = [1.0, 2.0, 3.0, 4.0, 1.0, 2.0, 3.0]に対して、
その softmax を計算するコードは、次のようになる。

```
>>> import numpy as np
>>> a = [1.0, 2.0, 3.0, 4.0, 1.0, 2.0, 3.0]
>>> np.exp(a) / np.sum(np.exp(a))
array([0.02364054, 0.06426166, 0.1746813, 0.474833, 0.02364054,
       0.06426166, 0.1746813])
```



softmax 関数の性質

- softmax関数は、それが適用される層のノードの softmax関数の値をすべて足し合わせると1になるという性質を持つ。それは、softmaxの出力が「確率」であることを考えればすぐわかる。
- 例えば、先の例で、ある画像が3である確率が 0.75であったとしよう。そのことは、その画像が3以外の数字である確率は、 $1 - 0.75 = 0.25$ だということである。3以外のノードの確率(softmaxの出力の値)を全部足しても、0.25にしかならないということである。
- 次のこともわかる。あるノードのsoftmaxの値が上がれば、他のノードの値は下がる。あるノードの値を下げれば、他のノードの値は上がる。全部足して1になるように、値は変化する。



One-Hot-Vector

- 先のsoftmax関数の出力例の、一番上のノードは、「この画像は数字nである」という、ニューラル・ネットワークの判断を表しているの。もちろん、softmaxの値が一番大きいところ選ばれて、そこに一つだけに1が入っていて、残りはすべて0になっている。
- こうした一つだけに1が、残りすべてが0のベクトルを、One-Hot-Vector と呼ぶ。
- One-Hot-Vectorは、クラス分けの出力に現れるだけではない。ニューラル・ネットワークを「学習」させるための訓練用データは、画像と一緒に「この画像は数字nである」という「正解」の情報を持っていなければならない。訓練用データは、画像と正解を示すOne-Hot-Vectorの二種類のデータから構成されている。



表記を見直す
ちょっと数学する



振り返りと表現の見直し

ここでは、二つの点でこれまでの表記を見直す。

- ニューラル・ネットワーク表記の見直し。
- ニューロンの状態を表す次の式の見直し。

$$W \cdot X^T + b^T$$


これらは、Googleのフレームワーク TensorFlowの表記に準じたものである。

ここでは、二つの点でこれまでの表記を見直す。

- ニューラル・ネットワーク表記の見直し。
- ニューロンの状態を表す次の式の見直し。
 $W \cdot X^T + b^T$

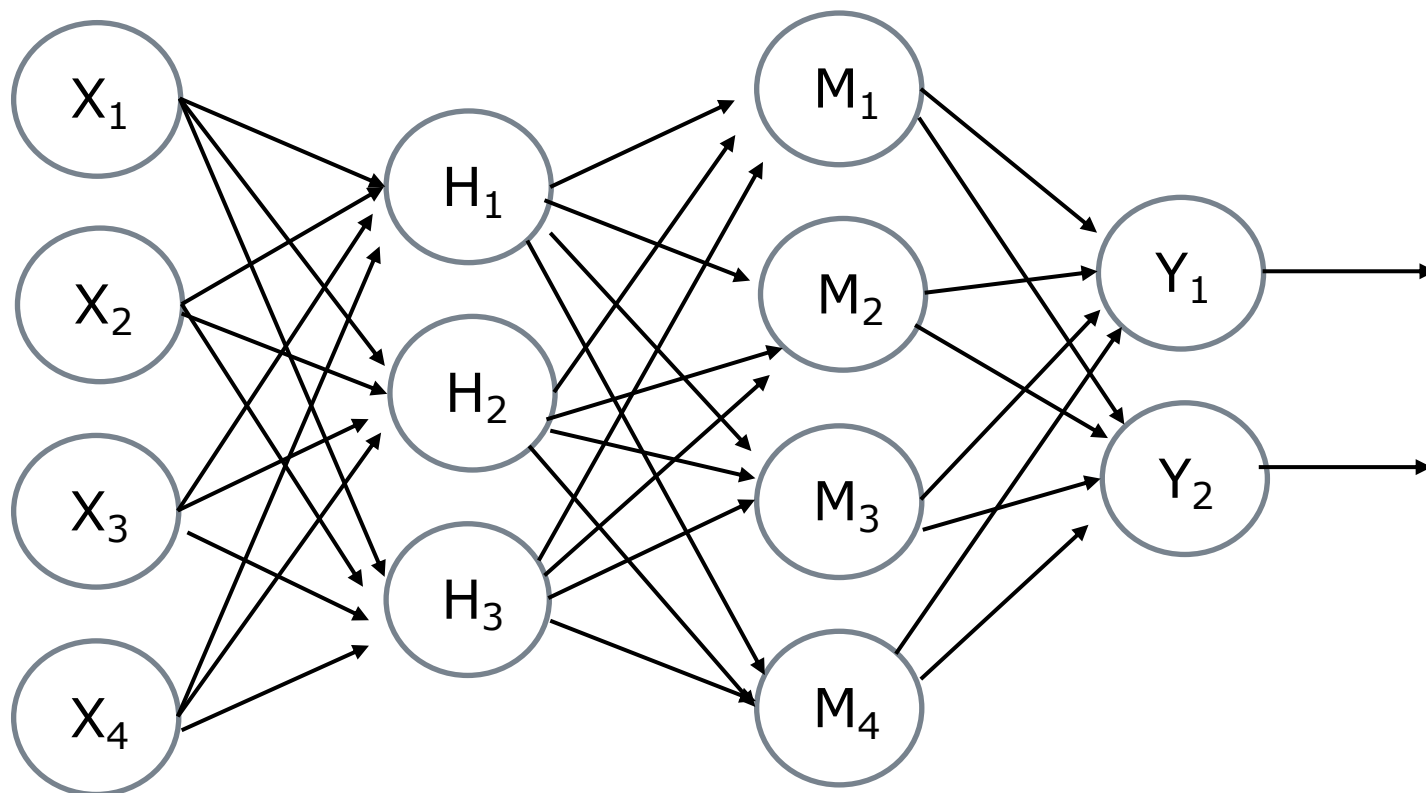
これらは、Googleのフレームワーク TensorFlowの表記に準じたものである。



ニューラル・ネットワーク表記の見直し 

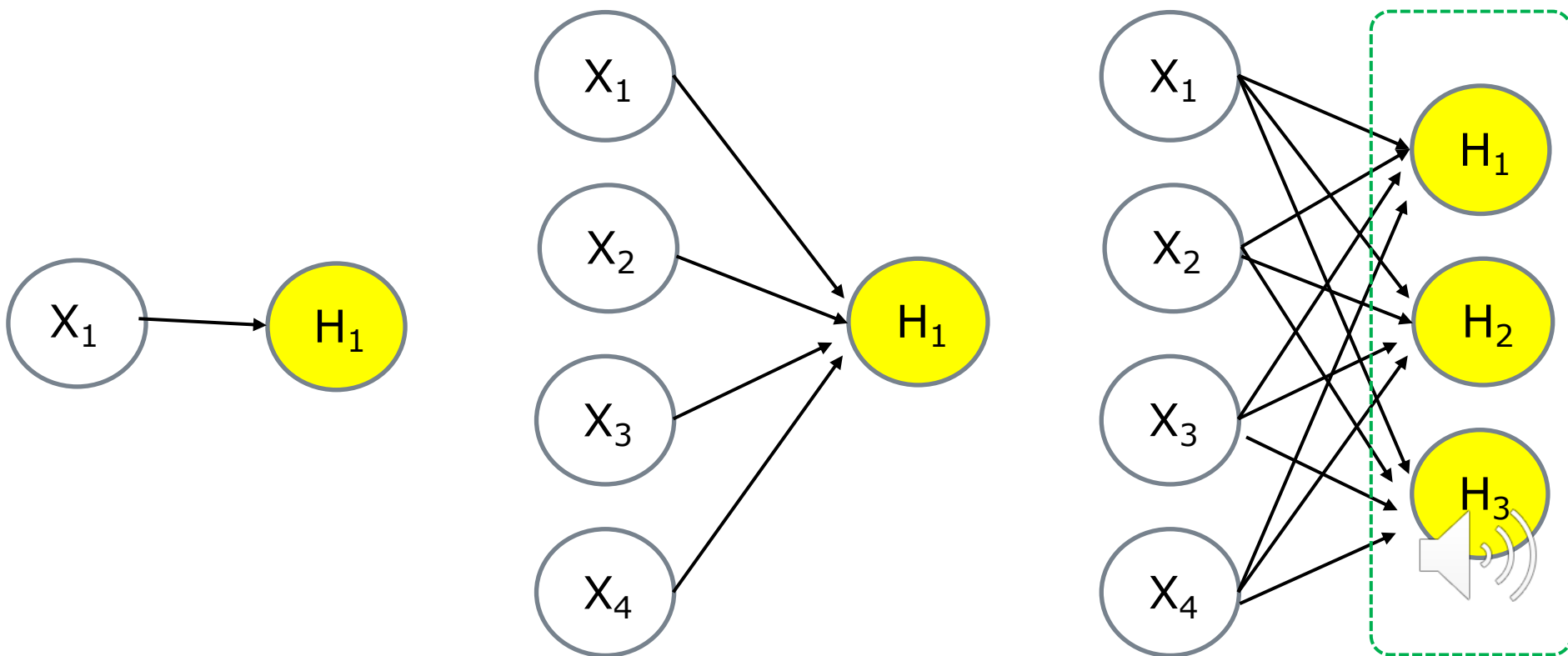
ニューラル・ネットワーク表記の見直し。 これまでの表記

これまでは、ニューラル・ネットワークを、ニューロンを丸で表し、ニューロンとニューロンの接続を直線で表してきた。

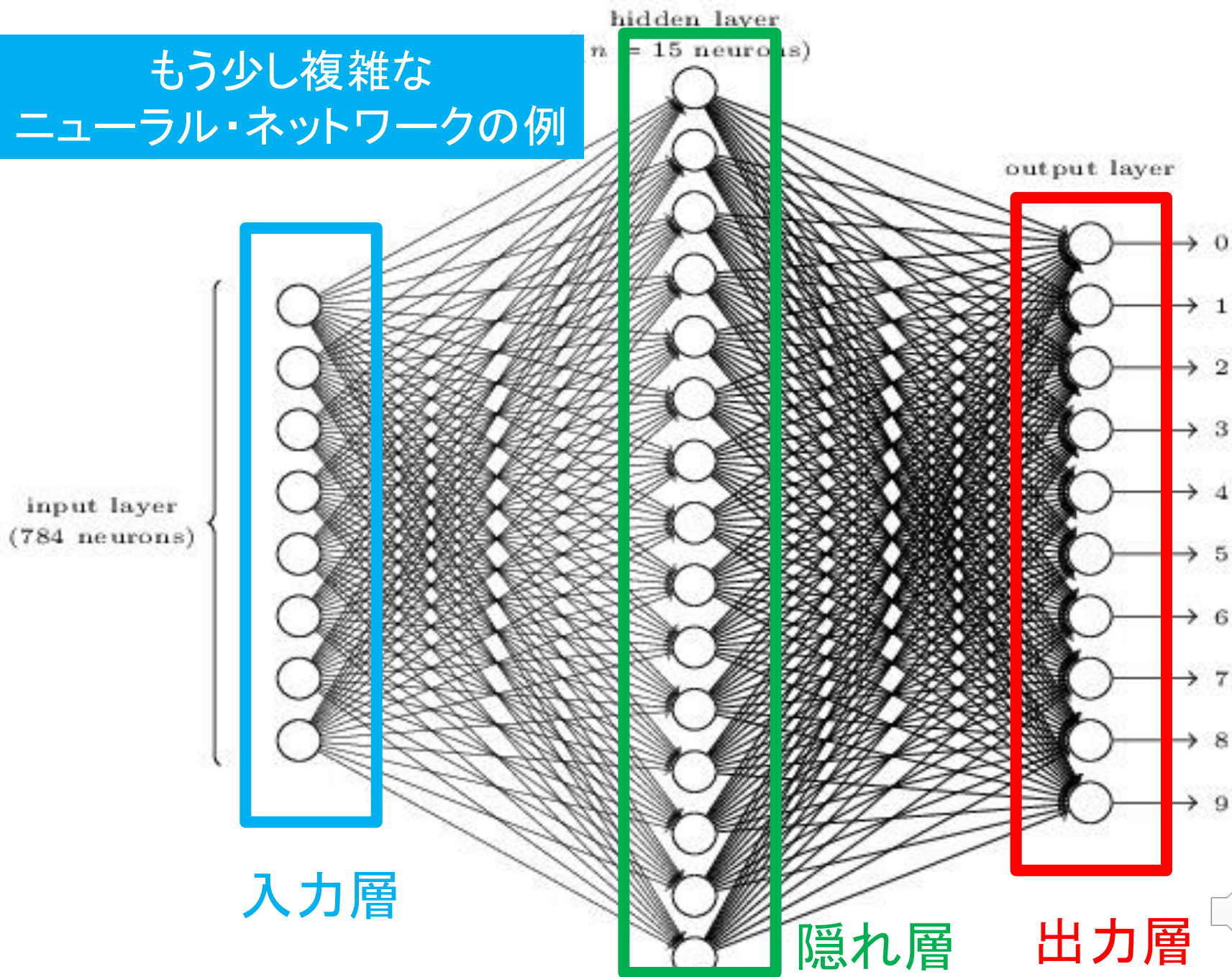


これまで見てきたニューラル・ネットワークは、 単純なものである

これまで見てきたネットワークは、次のように単純なものである。
複雑なものでも、入力層の後ろに、複数のニューロンからなる一つの層があるだけだった、



もう少し複雑な
ニューラル・ネットワークの例



入力層

隠れ層

出力層



もう少し複雑なネットワークを考える

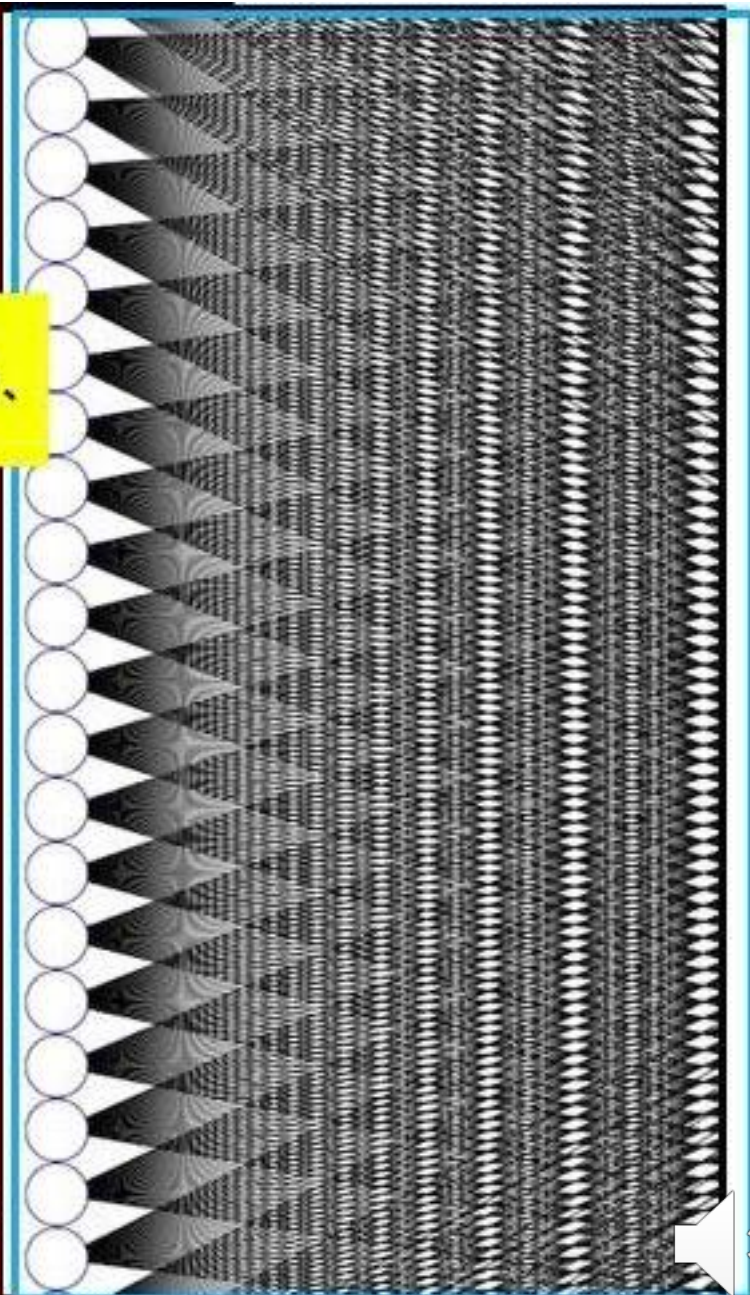
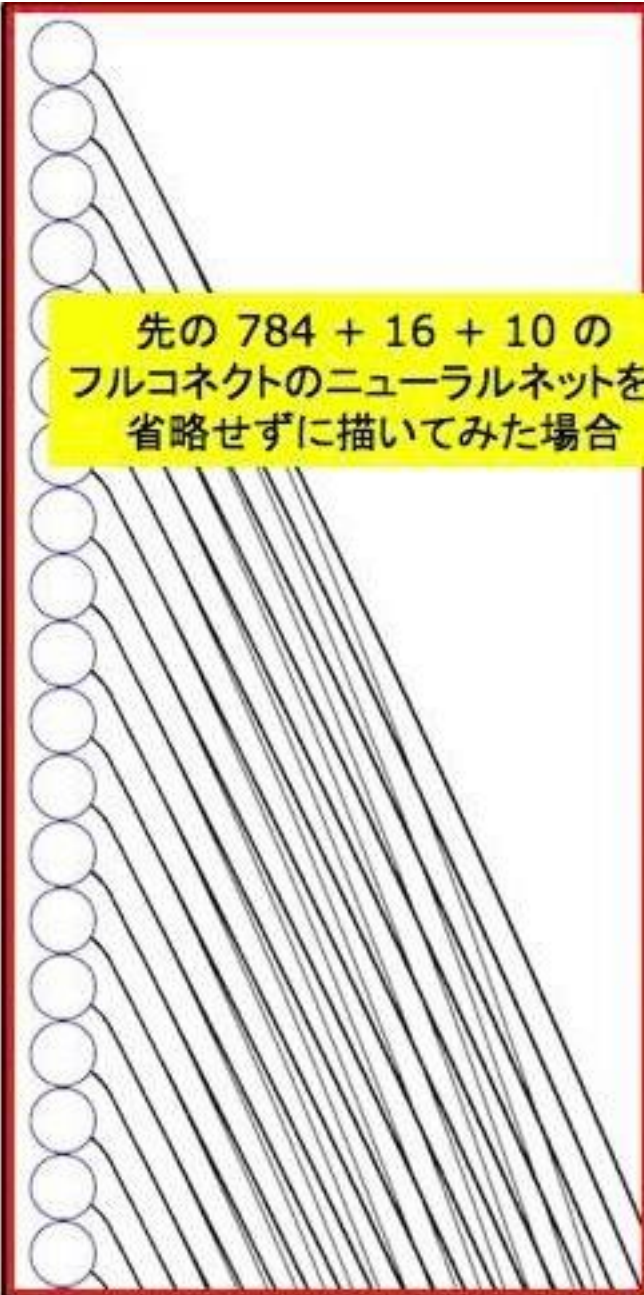
先のニューラル・ネットワークは、入力層の後ろに二つの層(隠れ層と出力層)を持っている。隠れ層は15個の、出力層は10個のニューロンから構成されている。

実は、この表示では入力層のニューロンは省略されている。実際には、入力層のニューロンは 784個あるというのだ。入力層のニューロンを含めてこれまでのスタイルでこのネットワークを表記したものの一部を次のページに示す。





先の $784 + 16 + 10$ のフルコネク트의ニューラルネットを、省略せずに描いてみた場合



ニューラル・ネットワークの表記をどう見直すか

入力層十二層からなる簡単なネットワークなのだが、これでは、さっぱりネットワークの全体像がわからない。

このネットワークは、わざと複雑に作ったものではない。入力層のニューロンの数784は 28×28 で、 28×28 のメッシュに書かれた0から9までの手書き数字を認識するために構成されたニューラル・ネットワークである。

ニューラル・ネットワークの表記の見直しが必要なことは、分かったと思うが、どう見直すかについては、次のニューロンの状態を表す式の見直しあとに、改めて述べたいと思う。





ニューロンの状態の式の見直し



ニューロンの状態の式の見直し

$$Y = W \cdot X^T + b^T$$

これまでは、ニューロンの状態を、式 $Y = W \cdot X^T + b^T$ で表現してきた。ここで、 X も b も行ベクトルである。 X^T も b^T も列ベクトルなので、ここではニューロンの状態 Y は、列ベクトル表されている。

ここでの見直しの中心は、ニューロンの状態も、行ベクトルで表すことである。

結論を先に言えば、

列ベクトル表示 $Y = W \cdot X^T + b^T$ は、

行ベクトル表示 $Y' = X \cdot W' + b$ に変換できる。

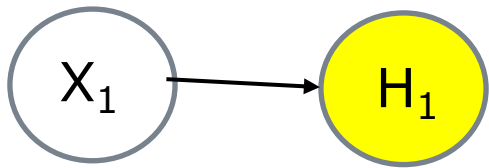
ここでは、そのことを見ていこう。



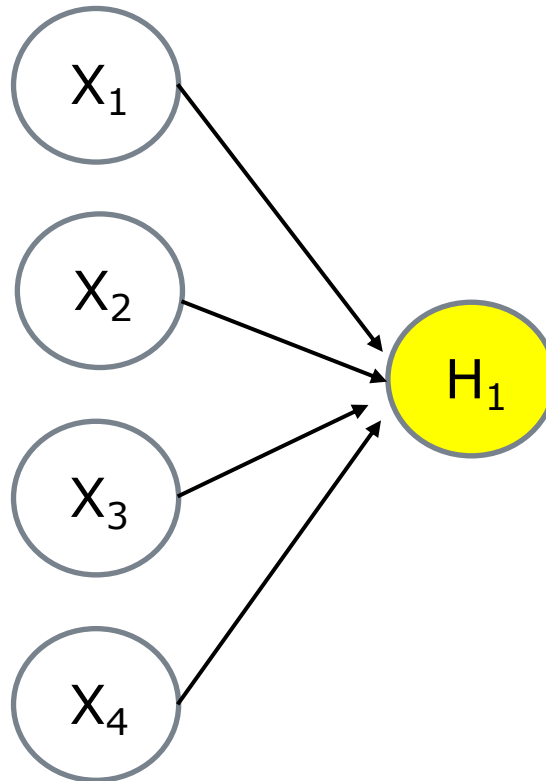
振り返り

ニューロンの状態の式 $Y = W \cdot X^T + b^T$

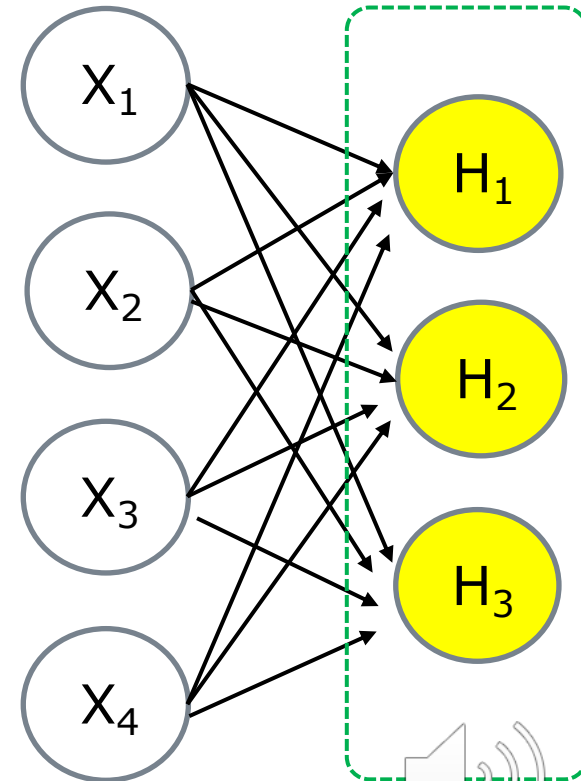
これまでは、ニューロンの状態を、式 $Y = W \cdot X^T + b^T$ で表現してきた。ここで、 X も b も行ベクトルである



$$Y = W \cdot X^T + b^T$$

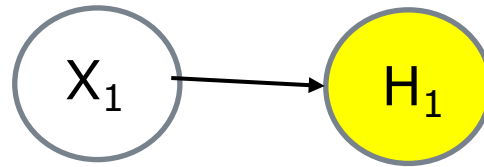


$$Y = W \cdot X^T + b^T$$



$$Y = W \cdot X^T + b^T$$

$Y = W \cdot X^T + b^T$ が表すもの



W, X, b は、数値 (スカラー)

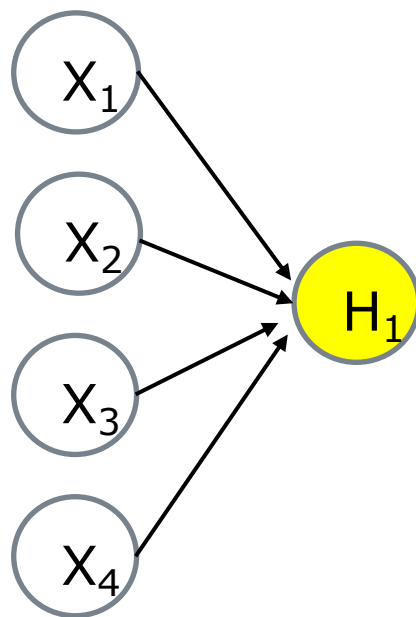
- $Y=y, X=x, W=a, b=b$ なら、この式は、
 $y = ax + b$ という、式を表す。

$Y = W \cdot X^T + b^T$ という式で

↑
スカラー W とスカラー X^T の積



$Y = W \cdot X^T + b^T$ が表すもの



W , X は行ベクトル、 b はスカラー

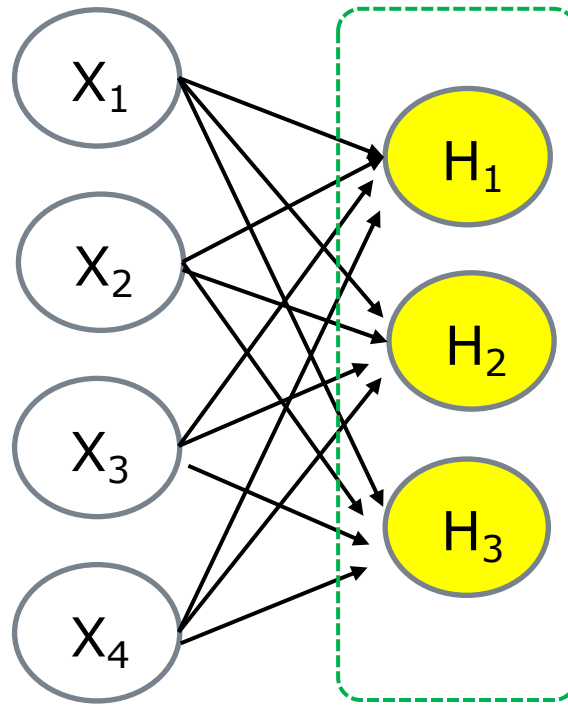
- $X = [X_1, X_2, X_3, X_4]$, $W = [W_1, W_2, W_3, W_4]$ なら
 $Y = W \cdot X^T + b$ という式を表す。



行ベクトル W と列ベクトル X^T の内積



$Y = W \cdot X^T + b^T$ が表すもの



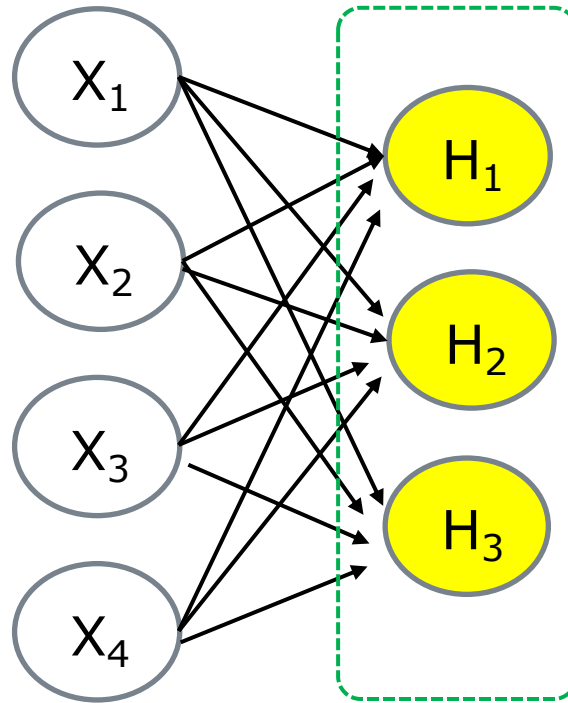
$Y = W \cdot X^T + b^T$ という式で

↑
行列 W と列ベクトル X^T の積



$$Y = W \cdot X^T + b^T \text{ が表すもの}$$

入力 $X = [X_1, X_2, X_3, X_4]$



重み $W_1 = [W_{11}, W_{12}, W_{13}, W_{14}]$
バイアス b_1

重み $W_2 = [W_{21}, W_{22}, W_{23}, W_{24}]$
バイアス b_2

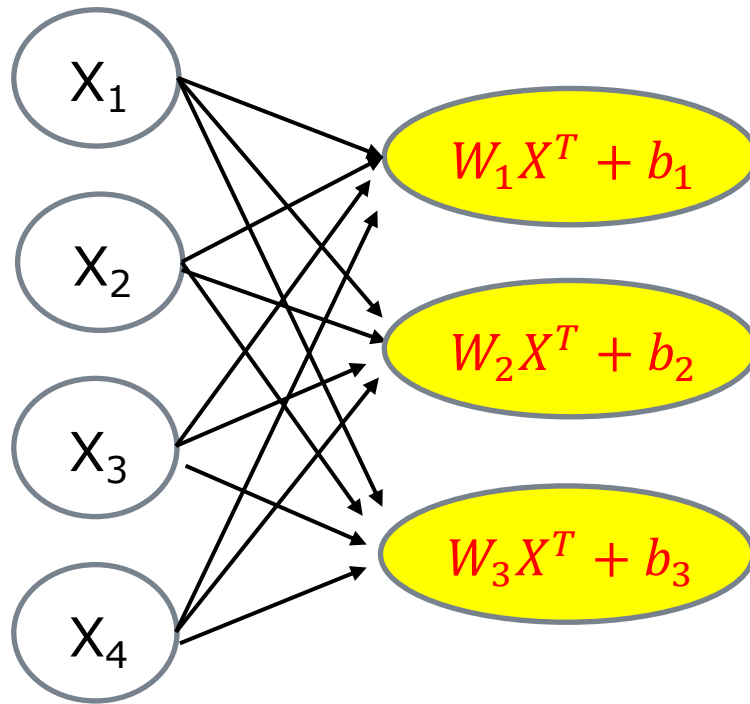
重み $W_3 = [W_{31}, W_{32}, W_{33}, W_{34}]$
バイアス b_3

W は行列, X は行ベクトル, b は行ベクトル



$$Y = W \cdot X^T + b^T \text{ が表すもの}$$

入力 $X = [X_1, X_2, X_3, X_4]$



重み $W_1 = [W_{11}, W_{12}, W_{13}, W_{14}]$
バイアス b_1

重み $W_2 = [W_{21}, W_{22}, W_{23}, W_{24}]$
バイアス b_2

重み $W_3 = [W_{31}, W_{32}, W_{33}, W_{34}]$
バイアス b_3

W は行列, X は行ベクトル、 $b = (b_1, b_2, b_3)$ は行ベクトル



重みの行ベクトル W_i から 重みの行列 W を作る

重みの行列 W は、重みの行ベクトル W_i から、次のように構成される。
ここで n はニューロンの数、 m は入力の数である。

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1m} \\ W_{21} & W_{22} & \dots & W_{2m} \\ & & \vdots & \\ W_{n1} & W_{n2} & \dots & W_{nm} \end{bmatrix}$$



重みの行ベクトル W_i から 重みの行列 W を作る

入力の行ベクトルを $X = [X_1 \ X_2 \ \dots \ X_m]$ 、
バイアスの行ベクトルを $b = [b_1 \ b_2 \ \dots \ b_n]$ とすると、

$$Y = WX^T + b^T = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} W_1X^T + b_1 \\ W_2X^T + b_2 \\ \vdots \\ W_nX^T + b_n \end{bmatrix}$$

これは、次の式に等しい。

$$\begin{bmatrix} W_{11} & W_{12} & \dots & W_{1m} \\ W_{21} & W_{22} & \dots & W_{2m} \\ \vdots & & & \\ W_{n1} & W_{n2} & \dots & W_{nm} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = WX^T + b^T$$





ちょっと数学



スカラーもベクトルも、行列とみなせる

m行n列の行列を $m \times n$ の行列と言う。この時

- スカラーは、 1×1 の行列
 - m次元の行ベクトルは、 $1 \times m$ の行列
 - n次元の列ベクトルは、 $n \times 1$ の行列
- とみなせる。
- Xをm次元の行ベクトルとすると、 X^T はm次元の列ベクトルである。(Xを $1 \times m$ の行列とすると、 X^T は $m \times 1$ の行列である)
 - Yをn次元の列ベクトルとすると、 Y^T はn次元の行ベクトルである。(Yを $m \times 1$ の行列とすると、 Y^T は $1 \times m$ の行列である)
 - Wを $m \times n$ の行列とすると、 W^T は $n \times m$ の行列である。



行列の積が可能な条件

$k \times l$ の行列 A と $m \times n$ の行列 B の積が可能なのは、 $l = m$ の場合だけである。この時、積 $C = MN$ は、 $k \times n$ の行列になる。

$$\text{行列 } A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1l} \\ A_{21} & A_{22} & \dots & A_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k1} & A_{k2} & \dots & A_{kl} \end{bmatrix} \text{ と行列 } B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \dots & B_{mn} \end{bmatrix}$$

$\underbrace{\hspace{10em}}_l$ $\underbrace{\hspace{10em}}_m$

の積が可能なのは、 $l = m$ の場合だけである。

A	B	$C = AB$
$k \times l$	$m \times n$?
$k \times l$	$l \times n$	$k \times n$

$\underbrace{\hspace{10em}}$



さまざまな行列の積

- n 次元の行ベクトル A と n 次元の列ベクトル B の積
 A は $1 \times n$ の行列、 B は $n \times 1$ 次元の行列
 AB は、 1×1 の行列。すなわち、スカラーである。
- $n \times m$ の行列 W と m 次元の列ベクトル V の積
 W は $n \times m$ の行列、 V は $m \times 1$ の行列。
 WV は、 $n \times 1$ の行列、すなわち n 次元の列ベクトルである。
- n 次元の行ベクトル X と $n \times m$ の行列 W の積
 X は $1 \times n$ の行列、 W は $n \times m$ の行列。
 XW は、 $1 \times m$ の行列、すなわち m 次元の行ベクトル。



トランスポーズTの性質

$$\begin{aligned}(A^T)^T &= A \\ (A + B)^T &= A^T + B^T \\ (AB)^T &= B^T A^T\end{aligned}$$

$Y = WX^T + b^T$ の時、

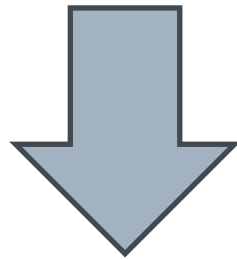
$$\begin{aligned}Y^T &= (WX^T + b^T)^T \\ &= (WX^T)^T + (b^T)^T \\ &= (X^T)^T W^T + b \\ &= XW^T + b\end{aligned}$$

これは、ネットワークの状態の行ベクトル表示である。



ネットワークの状態の行ベクトル表示

$$Y = WX^T + b^T = \begin{bmatrix} \boxed{W_{11} \ W_{12} \ \dots \ W_{1m}} \\ \boxed{W_{21} \ W_{22} \ \dots \ W_{2m}} \\ \vdots \\ \boxed{W_{n1} \ W_{n2} \ \dots \ W_{nm}} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

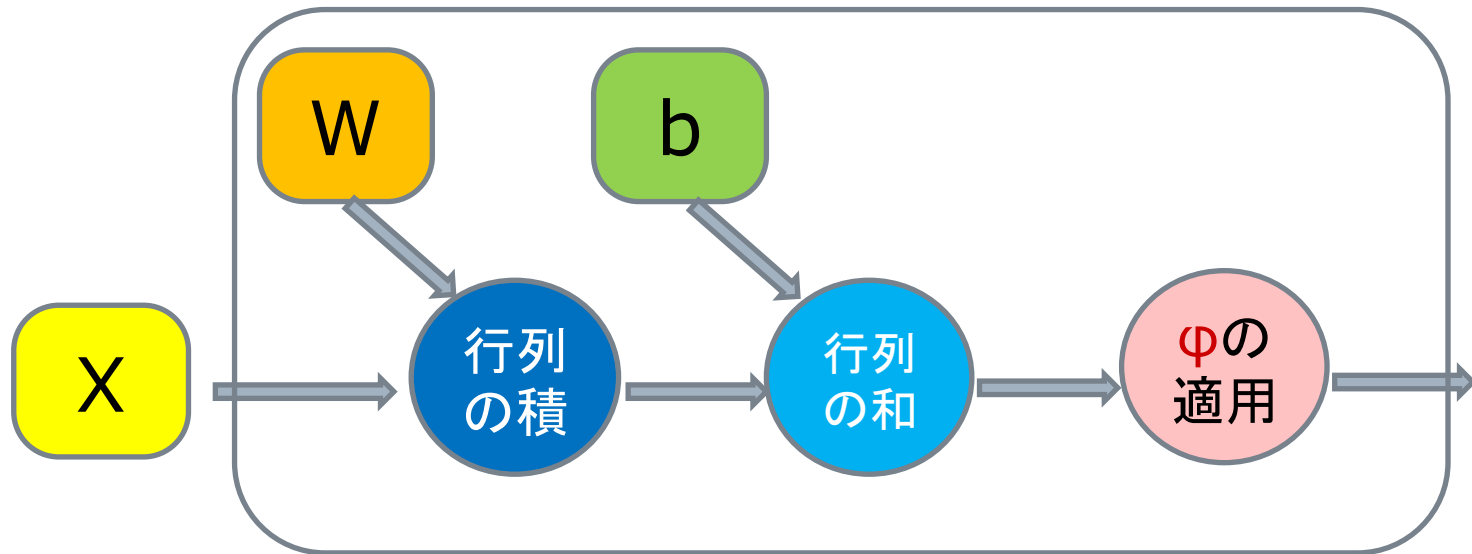


$$Y^T = XW^T + b = [X_1 \ X_2 \ \dots \ X_m] \begin{bmatrix} \boxed{W_{11}} & \boxed{W_{21}} & \dots & \boxed{W_{n1}} \\ \boxed{W_{12}} & \boxed{W_{22}} & \dots & \boxed{W_{n2}} \\ \vdots & \vdots & & \vdots \\ \boxed{W_{1m}} & \boxed{W_{2m}} & \dots & \boxed{W_{nm}} \end{bmatrix} + [b_1 \ b_2 \ \dots \ b_n]$$

$W_1 \quad W_2 \quad \dots \quad W_n$



一つの「層」のグラフでの表現



複数の層からなるネットワークを、
グラフで簡潔に表す





複数のニューロンからなる
一つの層を、グラフで表す



複数のニューロンからなる一つの層を、グラフで表す

前回見たニューラル・ネットワークの状態の行ベクトル表示 $Y=XW+b$ をベースに、新しいグラフによるネットワークの新しい表示を紹介する。

まずは、「複数のニューロンからなる一つの層」のグラフ表示を考える。



複数のニューロンからなる一つの層の状態

入力 $X = [X_1, X_2, X_3]$

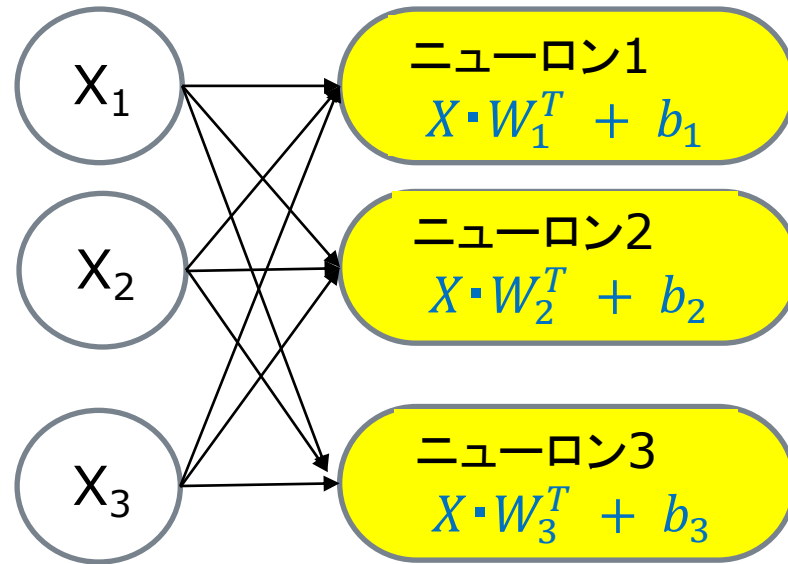
状態 $Y_i = X \cdot W_i^T + b_i$

重み $W_1 = [W_{11}, W_{12}, W_{13}]$
バイアス b_1

重み $W_2 = [W_{21}, W_{22}, W_{23}]$
バイアス b_2

重み $W_3 = [W_{31}, W_{32}, W_{33}]$
バイアス b_3

バイアス $b = [b_1 \ b_2 \ b_3]$



➡
$$Y = [X_1 \ X_2 \ X_3] \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \end{bmatrix} + [b_1 \ b_2 \ b_3]$$

複数のニューロンの状態は、 $X \cdot W + b$ の形で表せる。



Activatorの見直し

複数のニューロンからなる一つの層の出力

- SoftMaxは、n次元のベクトルを受け取って、n次元のベクトルで表される確率分布を返すActivator だった。
- ただ、先に見たSoftMax以外のActivator ϕ は、ニューロンの状態の値をニューロンの出力に変える、スカラーをスカラーに変える関数だった。
- Activator ϕ を、名前とスカラーに対する振る舞いは同じままで、SoftMaxと同じように、引数にn次元のベクトルを取り、n次元のベクトルを返すように拡張する。

$$\phi([v_1 \ v_2 \ \dots \ v_n]) = [\phi(v_1) \ \phi(v_2) \ \dots \ \phi(v_n)]$$

この時、

複数のニューロンの出力は、 $\phi(X \cdot W + b)$ の形で表せる。

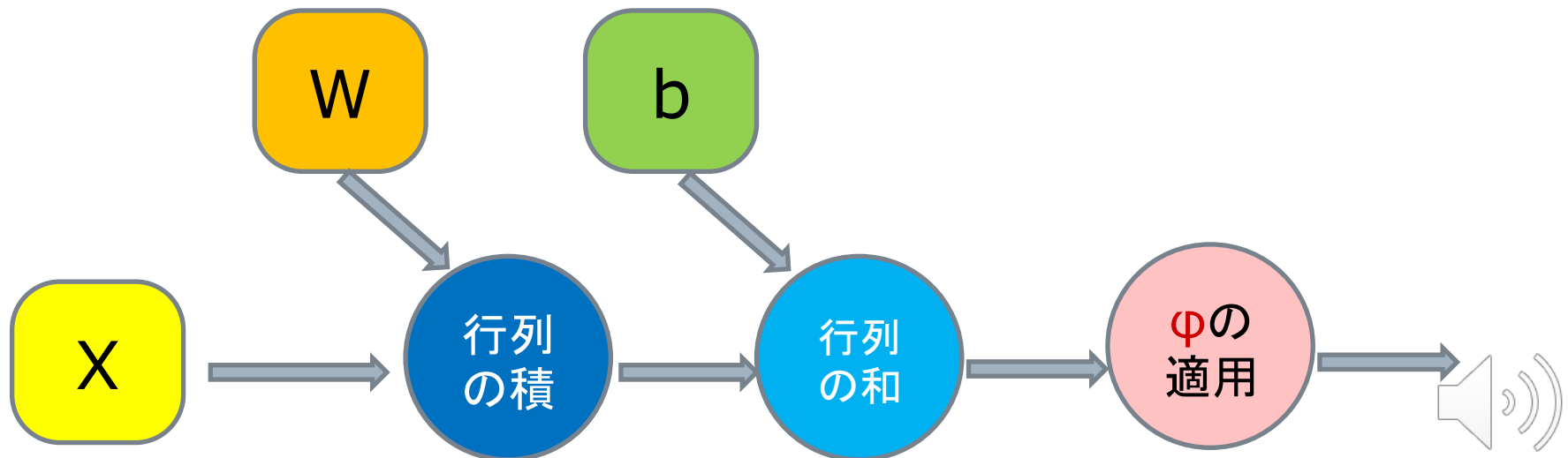


新しいグラフ

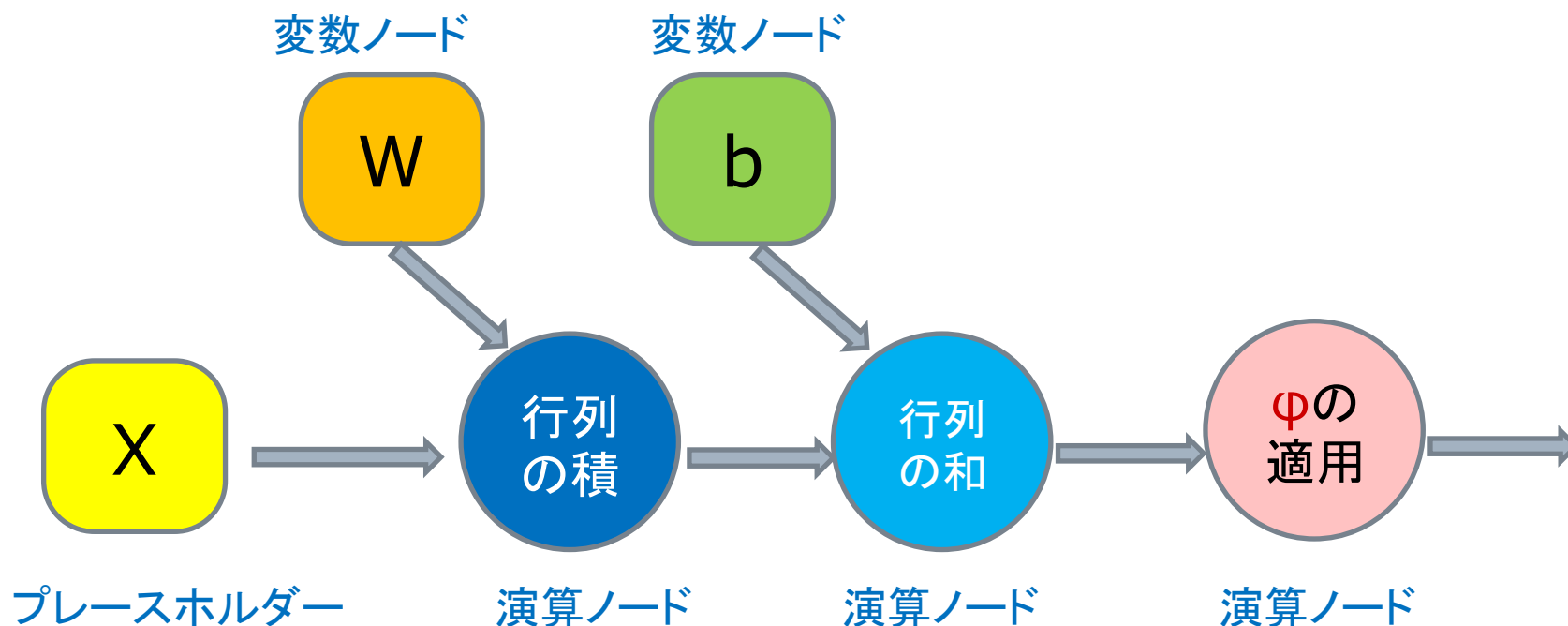
式そのものを図形として表現する

これまでは、ニューラル・ネットワークの表示は、ニューロンをノードとし、ニューロンとニューロンの接続を直線で表してきた。

新しい表示では、 $\varphi(X \cdot W + b)$ といった式そのものを、次のようなグラフで表すことにしよう。



こうした図形を「グラフ」と呼ぼう。
グラフは、「ノード」とノード同士を結ぶ「辺」からできている。



グラフのノードは、基本的には、演算を行う「演算ノード」である。
様々な演算が用意されている。演算を行わない「変数ノード」
「プレースホルダー・ノード」も存在する。

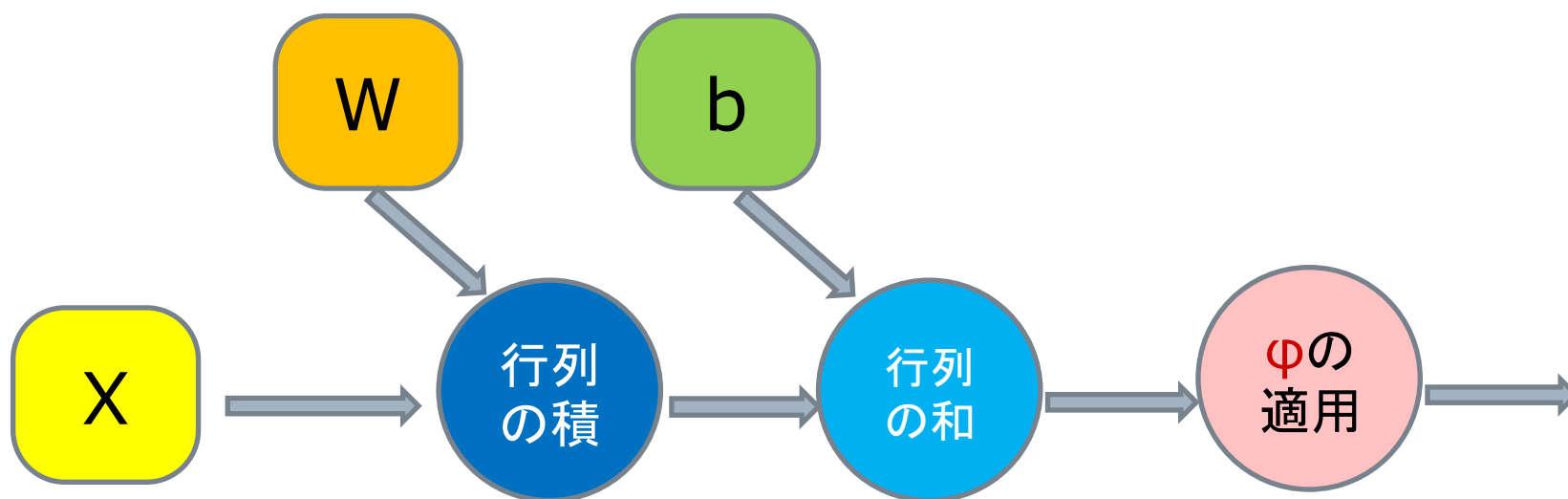


一つのグラフで、 さまざまなネットワークを表現する

さまざまといってもここで扱うのは、「複数のニューロンからなる一つの層」のネットワークである。



このグラフが有用なのは、一つと同じグラフで、様々な複雑なニューラル・ネットワークを表現できるからである。
(もちろん、 X, W, b のタイプは異なるのだが)

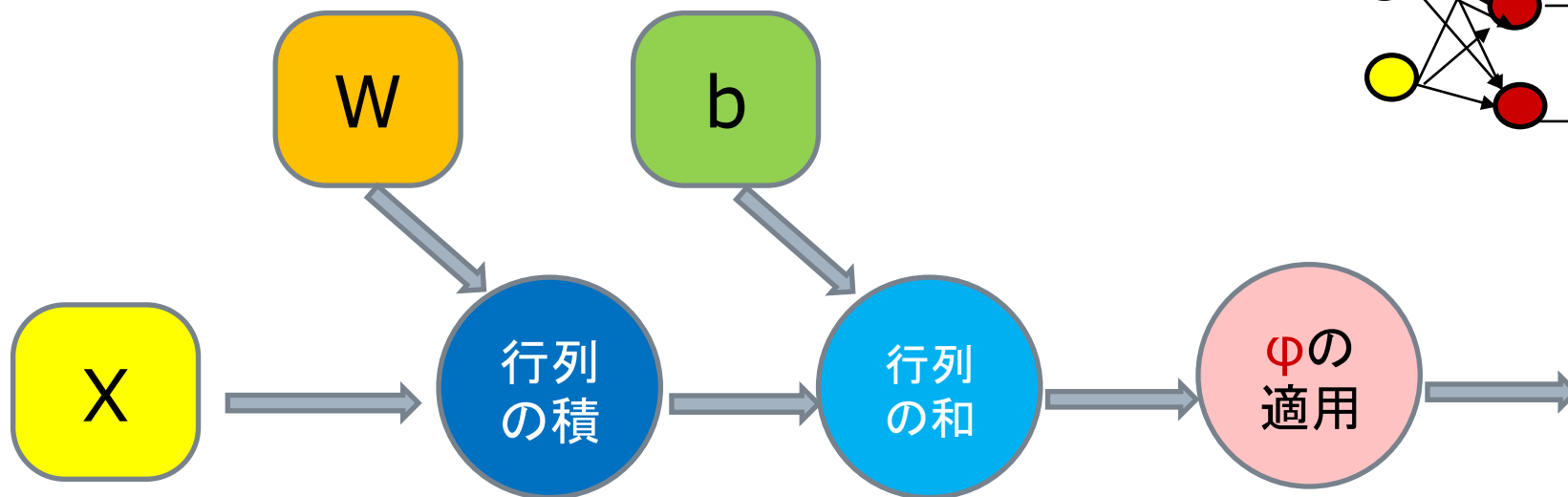
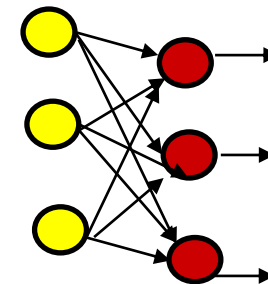


もっとも、そのことは、基本的には、このグラフの場合には、すでに見た $\varphi(X \cdot W + b)$ という数学的な抽象化に負っている。ただ、グラフの方が具体的に処理をイメージしやすい。

いくつかの例で考えてみよう。先に見たのは、
3つの入力を受け取り、3つの出力を返す、3つのニューロン
からなる「層」だった。

3行3列の行列

$[b_1, b_2, b_3]$

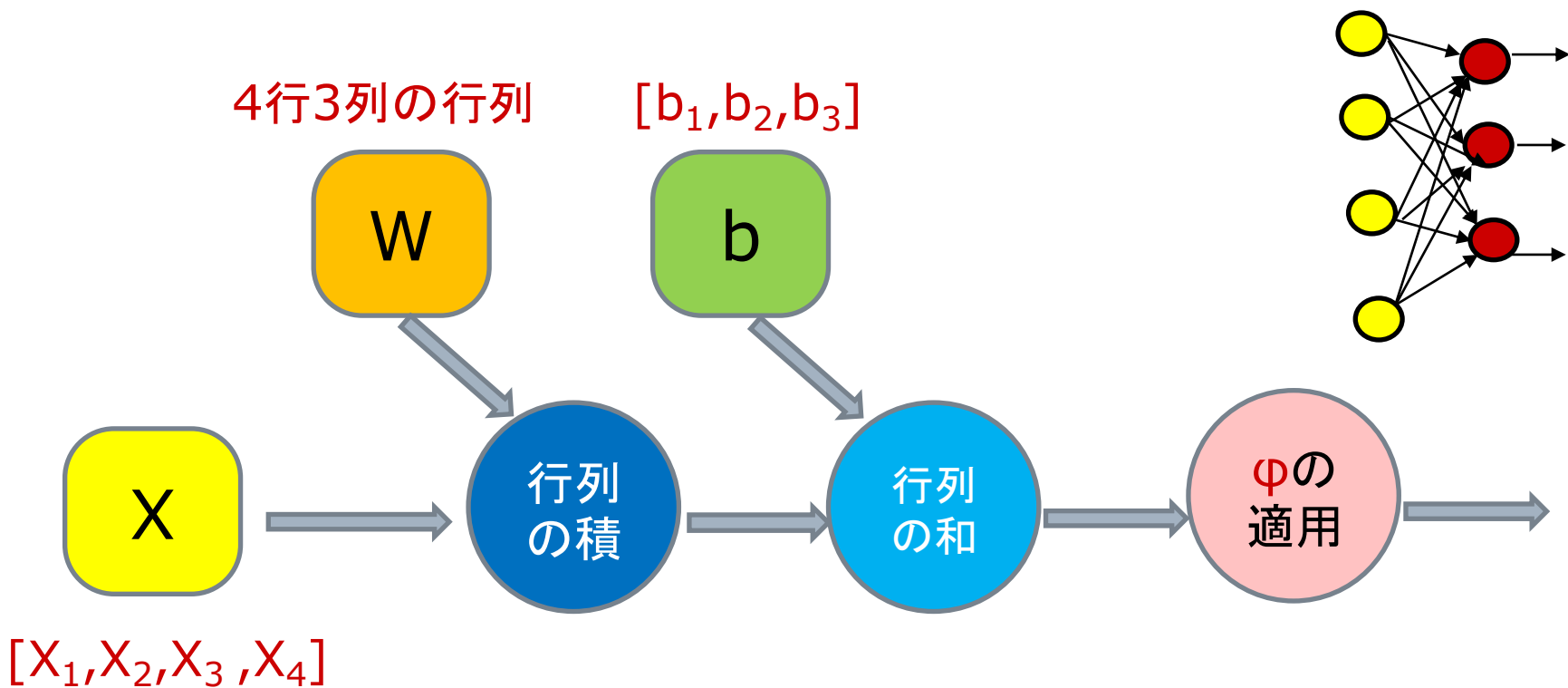


$[X_1, X_2, X_3]$

ある「層」の出力の数は、その層に含まれるニューロンの数に
等しい。



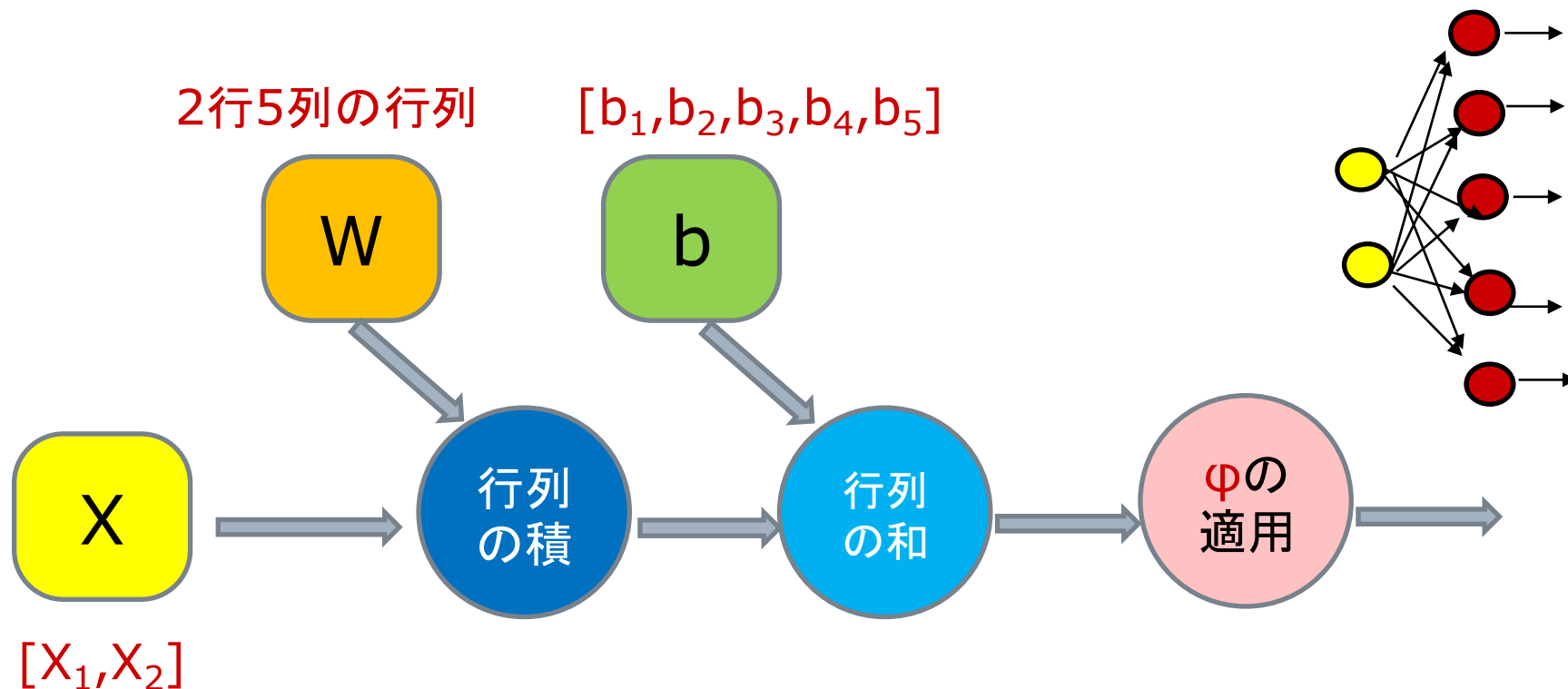
4つの入力を受け取り、3つの出力を返す、3つのニューロンからなる「層」を考えてみよう。これも、このグラフで表現できる。



このグラフと先のグラフは同じ形をしている。
ただし、ノード X , W の値は異なっている。




2つの入力を受け取り、5つの出力を返す、5つのニューロンからなる「層」を考えてみよう。これも、このグラフで表現できる。



一般に、 n 個の入力を受け取り、 m 個の出力を返すニューロンの「層」では、 X は n 次元の行ベクトル、 W は n 行 m 列の行列、 b は m 次元の行ベクトルである。





複数の層からなる
ニューラル・ネットワークをグラフで表す 

複数の層からなるニューラル・ネットワークをグラフで表す

もちろん、通常の表現でも、ニューラル・ネットワークはグラフとして表現されていた。ただ、新しいグラフは、それをさらに単純化する。

そうした単純化されたグラフ表現は、グラフを流れる「テンソル」というデータの捉え方とは、結びついている。

スカラーもベクトルも行列も、テンソルである。



Full Connect Network

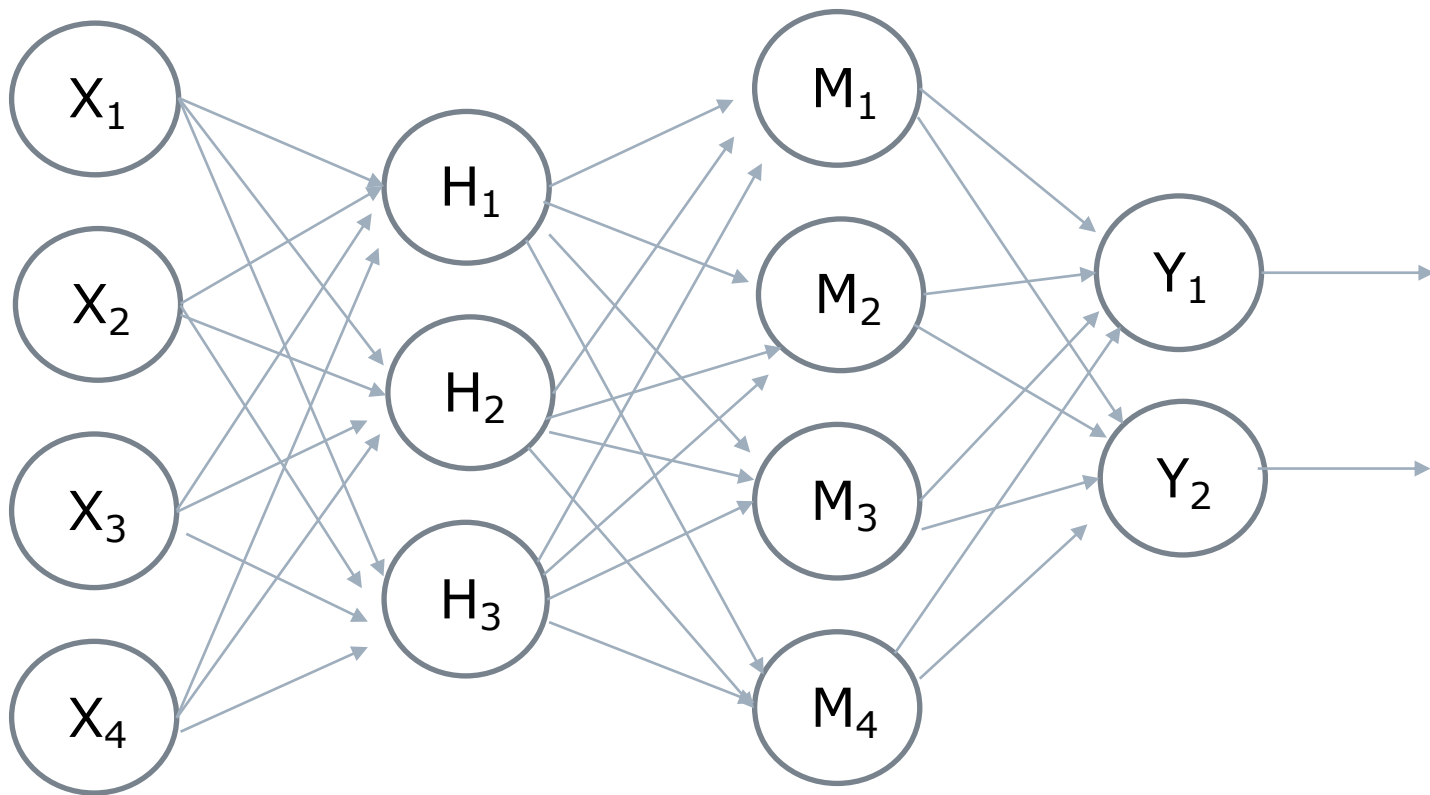
複数の層からなるネットワークで、向かい合った層のノードがすべて接続されているネットワークを、Full Connect Network という。

Full Connect Networkを、DNN(Deep Neural Network), Feed forward Network と呼ぶこともある。

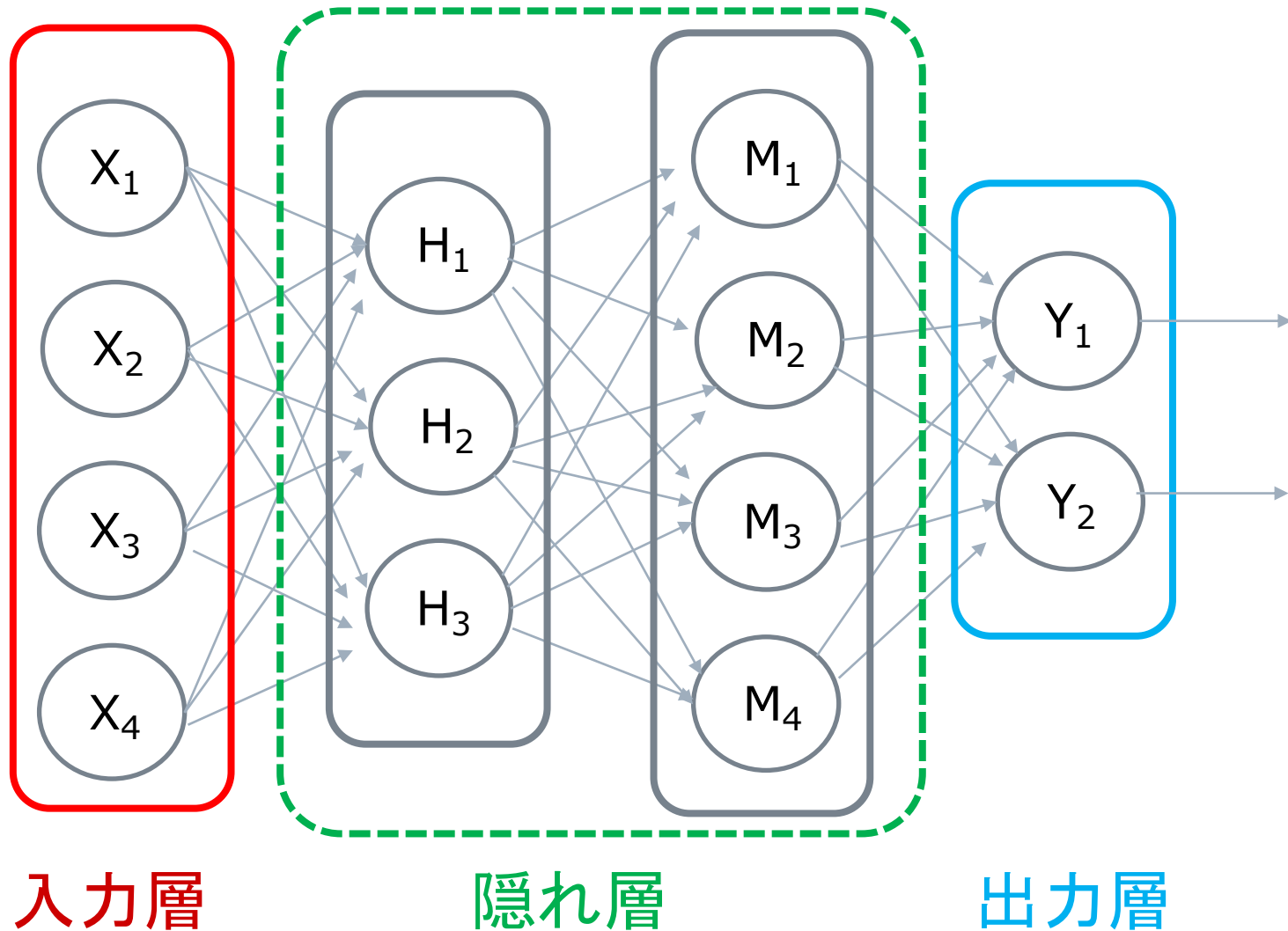
最も基本的なニューラル・ネットワークのパターンである。



Full Connectなニューラル・ネットワークの 旧来のグラフでの表現



Full Connectなニューラル・ネットワークの 旧来のグラフでの表現

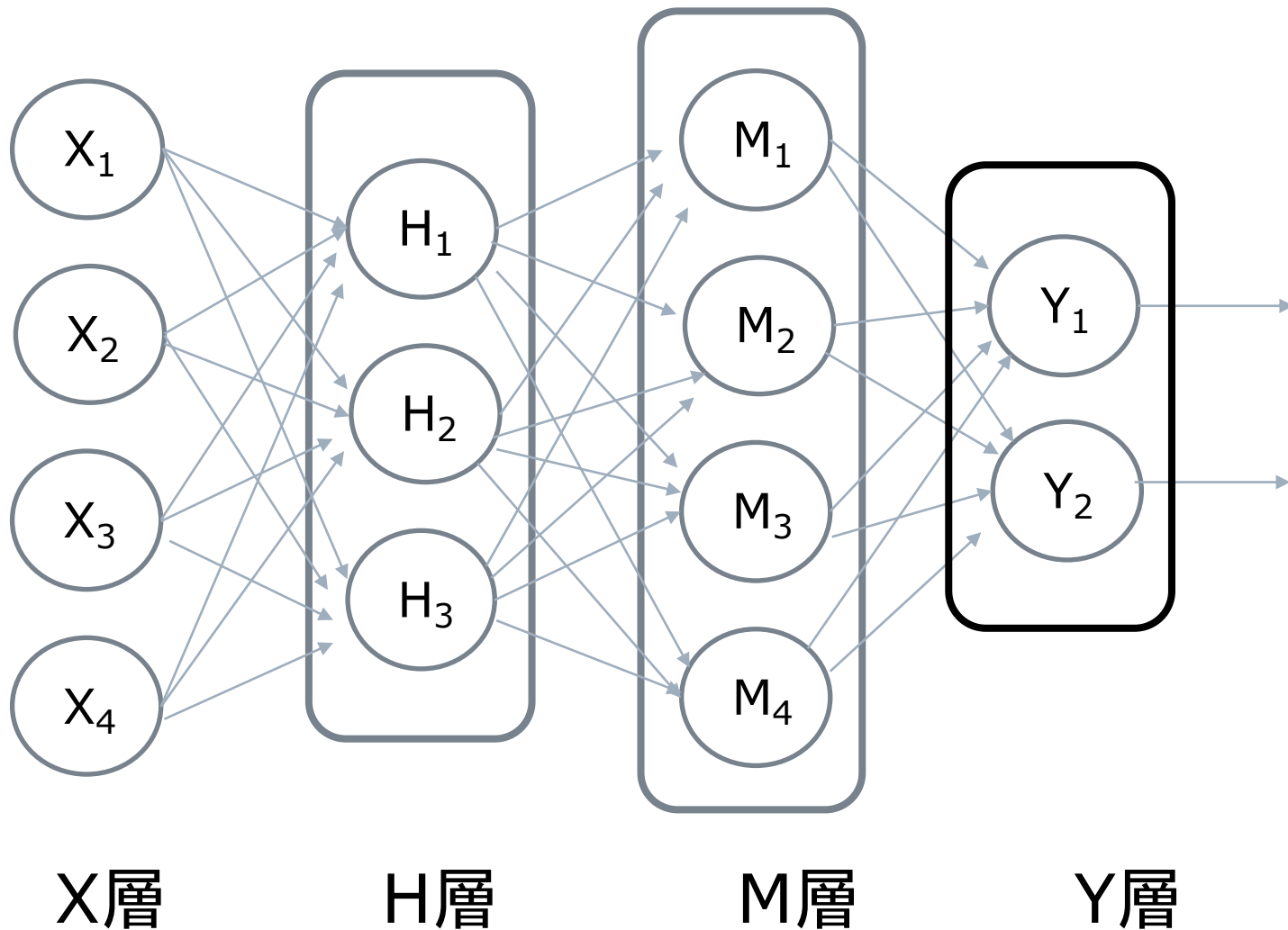


旧来のグラフ

H層の
重み W_H
バイアス b_H
活性化 ϕ_H

M層の
重み W_M
バイアス b_M
活性化 ϕ_M

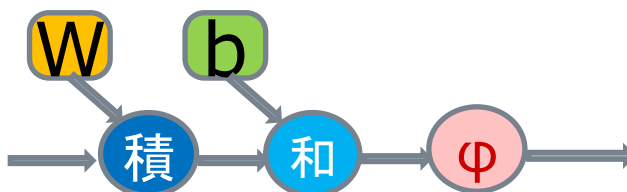
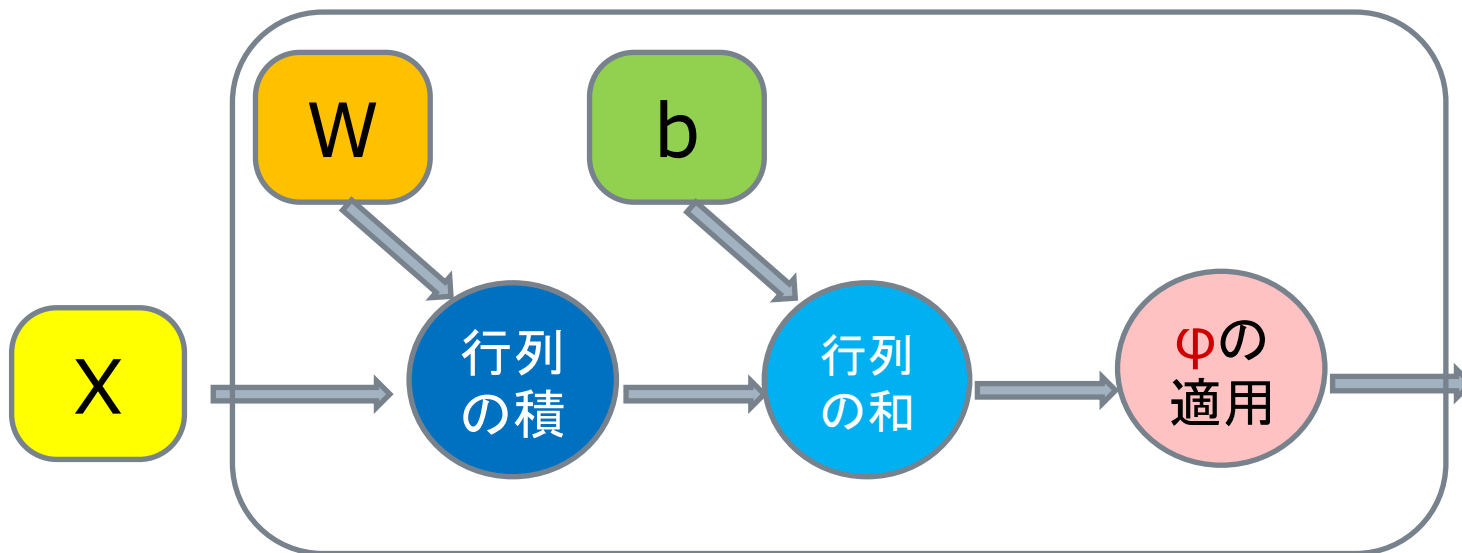
Y層の
重み W_Y
バイアス b_Y
活性化 ϕ_Y



Full Connect Networkの グラフでの表現



一つの「層」のグラフでの表現

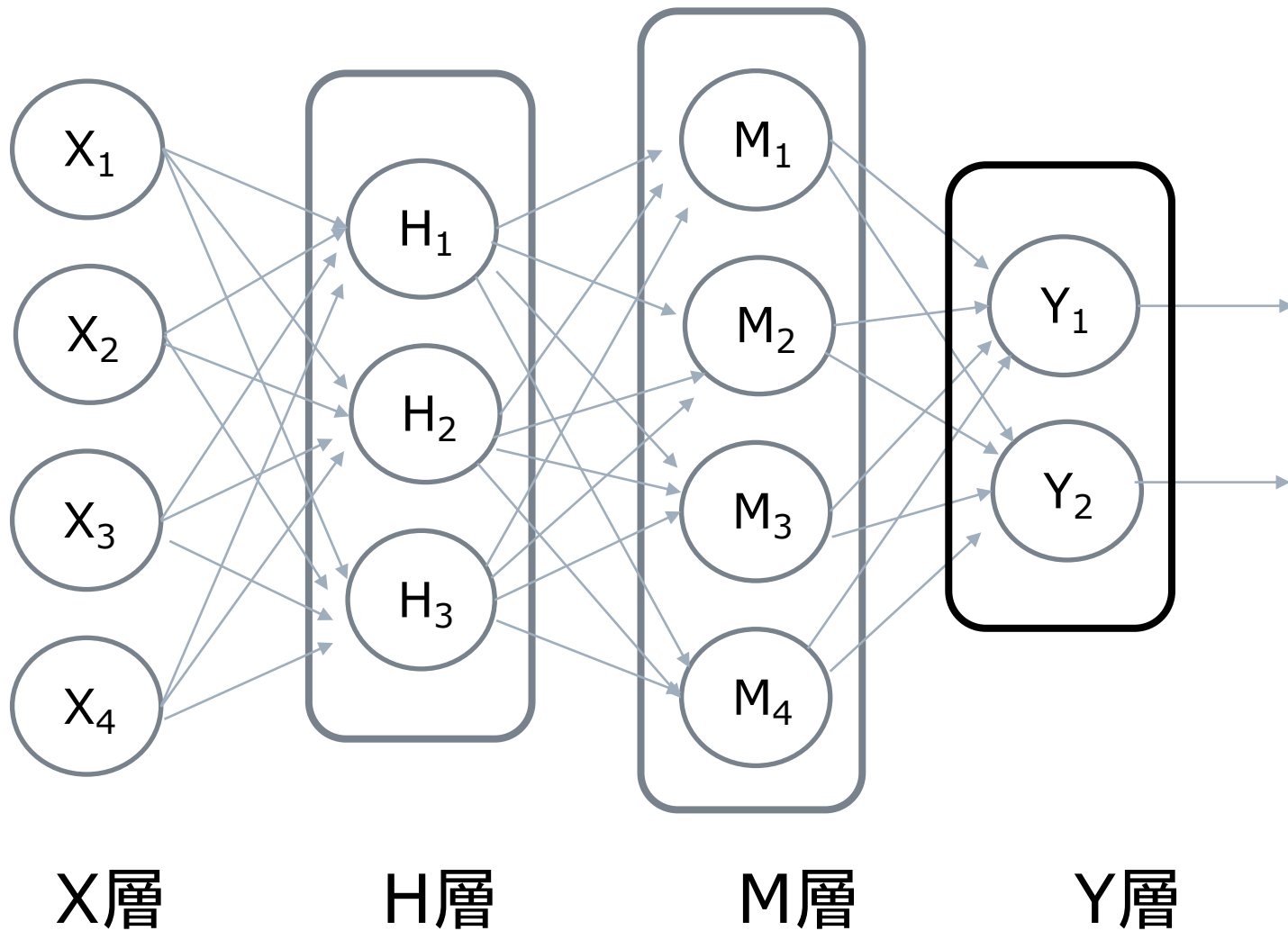


旧来のグラフ

H層の
重み W_H
バイアス b_H
活性化 ϕ_H

M層の
重み W_M
バイアス b_M
活性化 ϕ_M

Y層の
重み W_Y
バイアス b_Y
活性化 ϕ_Y



新しいグラフ

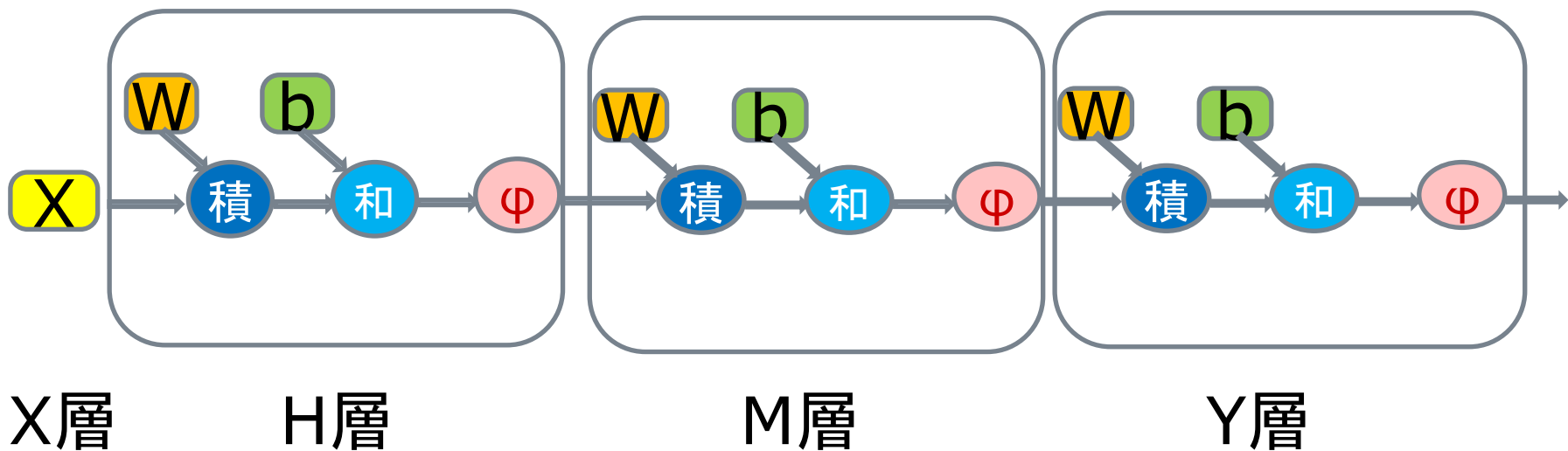
それぞれの層を表すグラフを
直列に結合すればいい

X層の
入力: 4
4次元

H層の
ニューロン数: 3
重み W : 4x3次元
バイアス b : 3次元

M層の
ニューロン数: 4
重み W : 3x4次元
バイアス b : 4次元

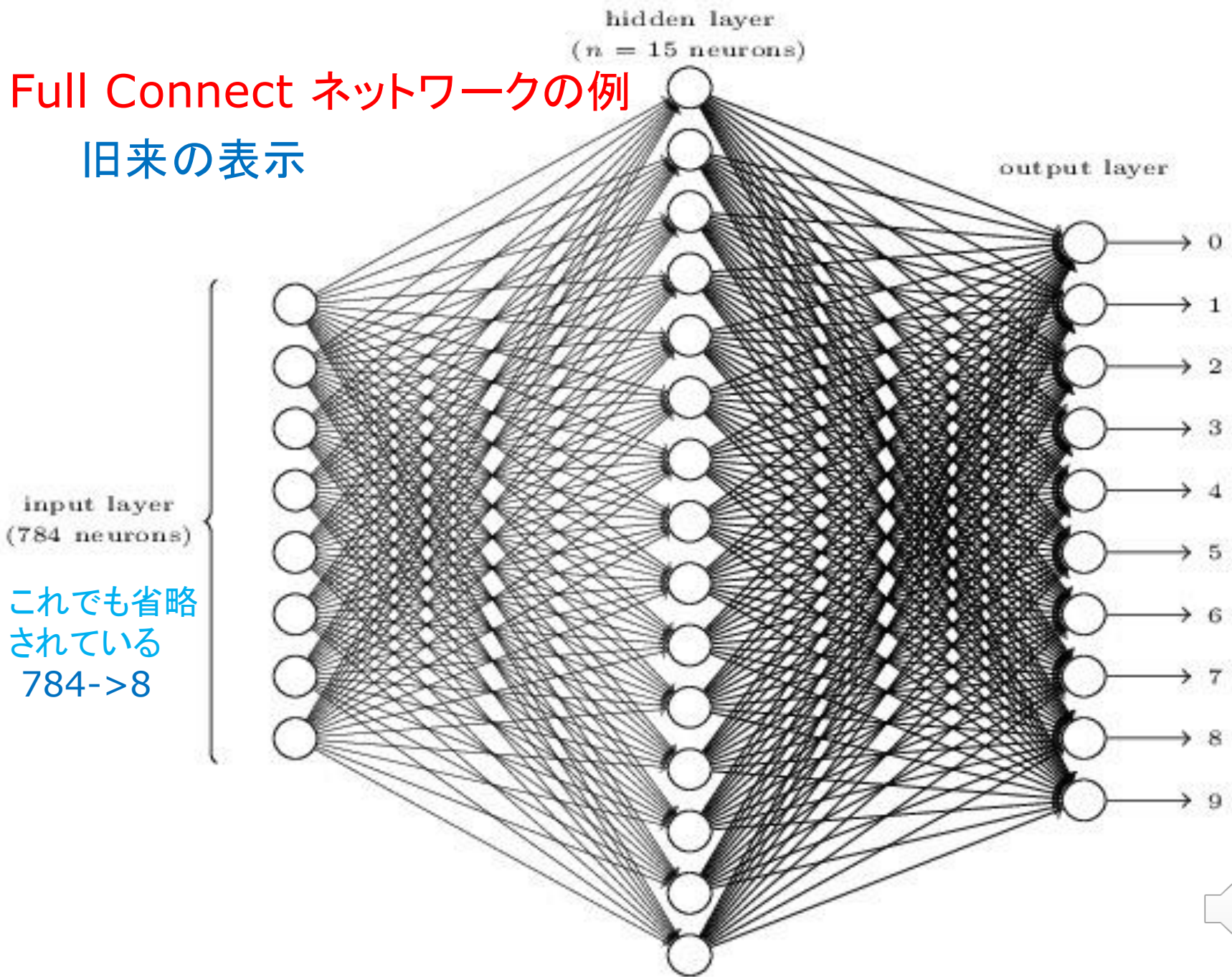
Y層の
ニューロン数: 2
重み W : 4x2次元
バイアス b : 2次元



こっこのグラフの方が、ずっとわかりやすい 

Full Connect ネットワークの例

旧来の表示

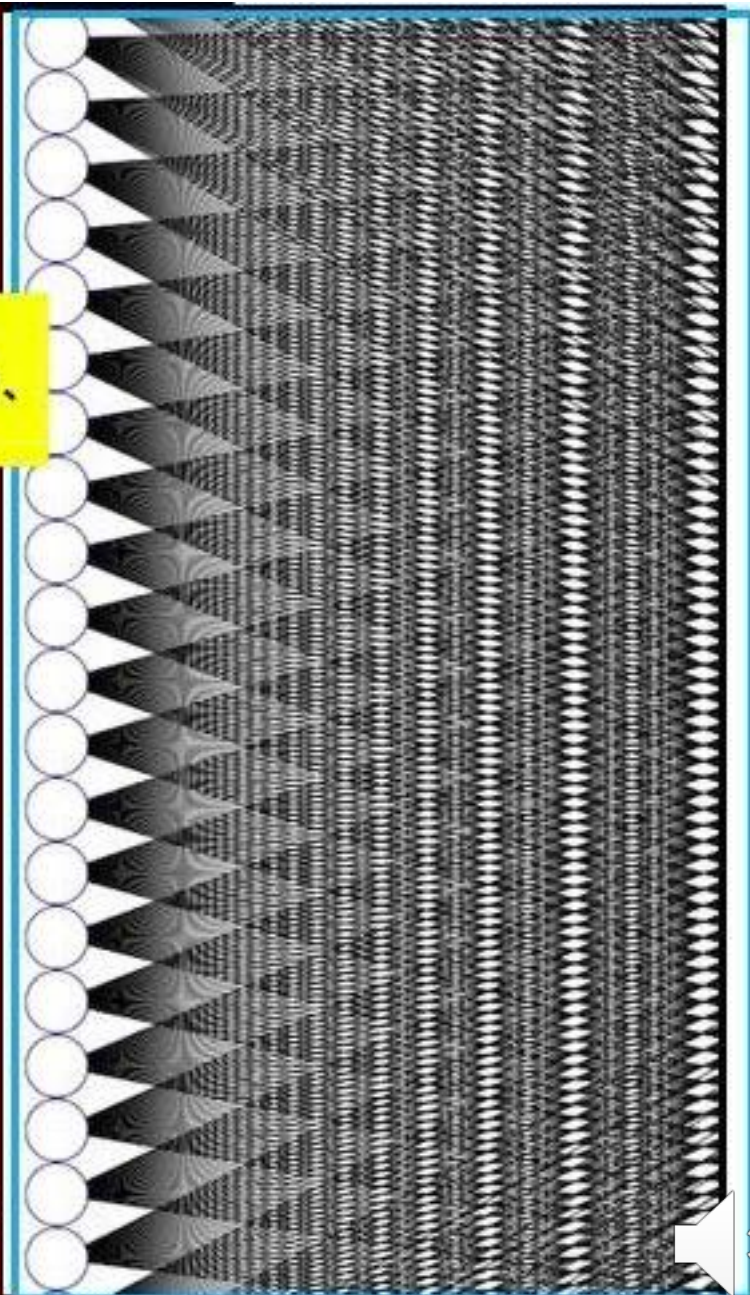
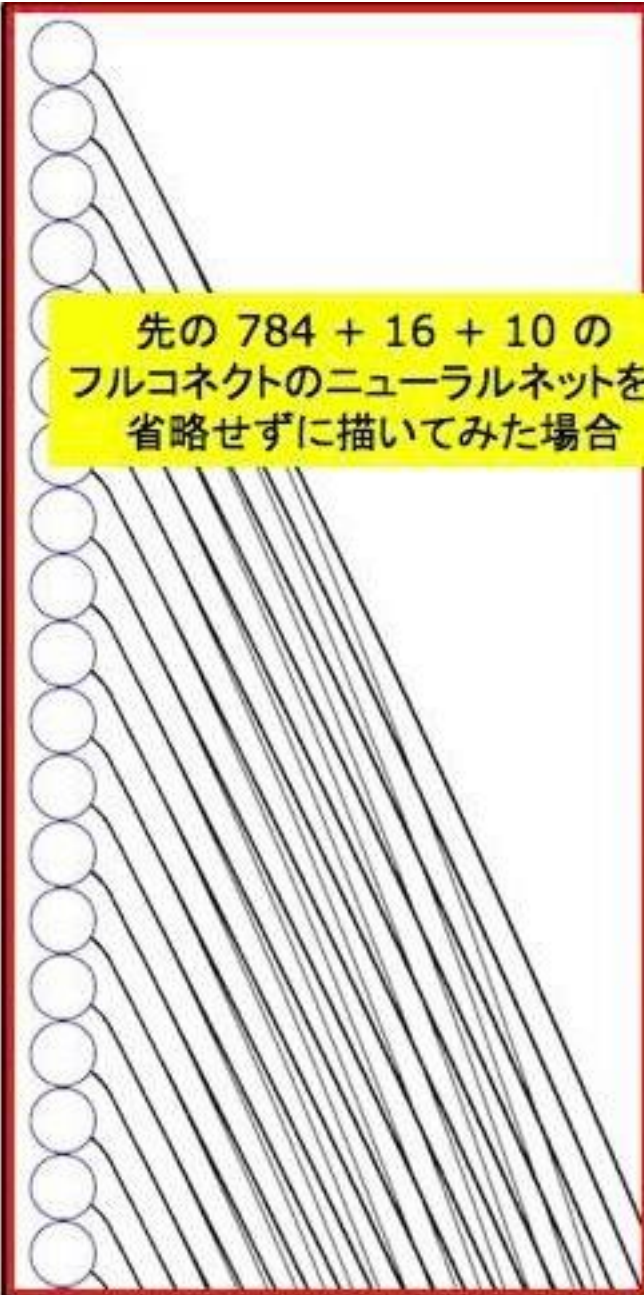


これでも省略
されている
784- \rightarrow 8





先の $784 + 16 + 10$ のフルコネクットのニューラルネットを、省略せずに描いてみた場合



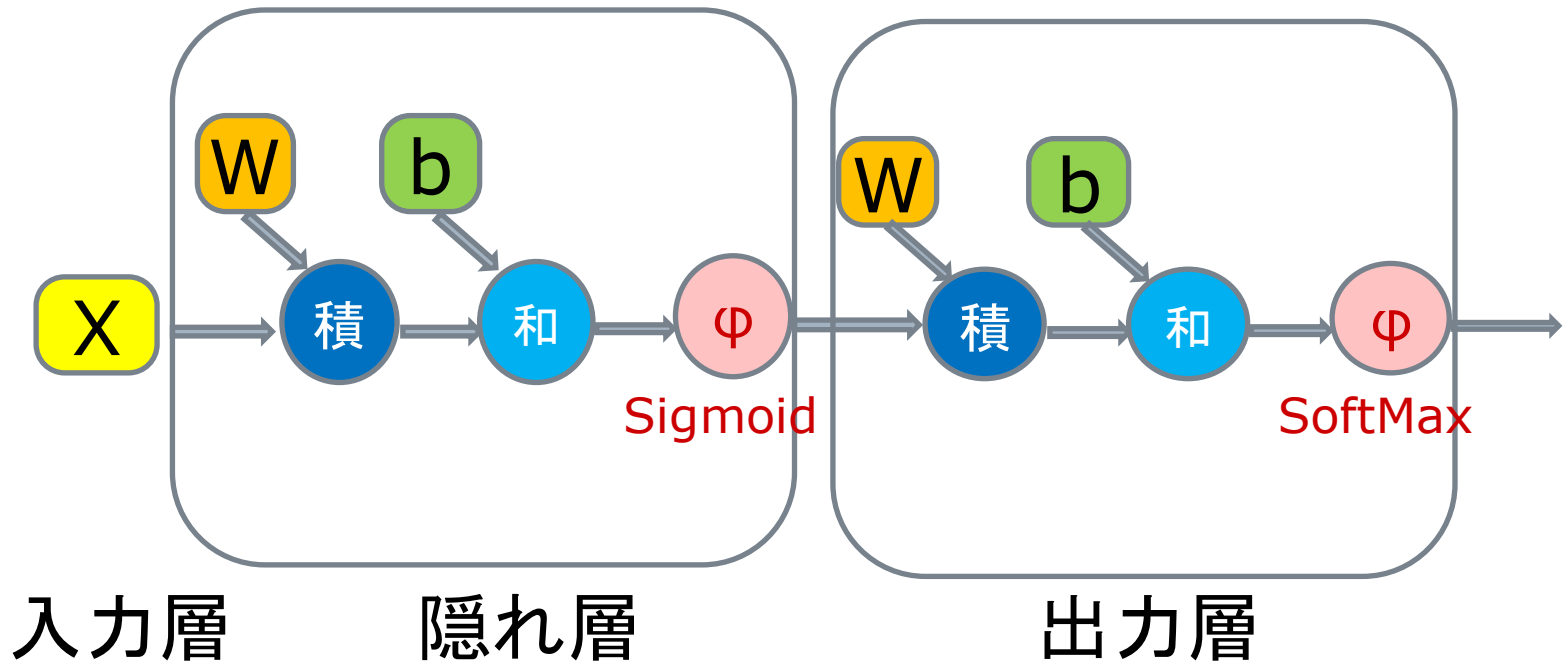
新しいグラフ

それぞれの層を表すグラフを
直列に結合すればいい

隠れ層の
入力: 784
784次元

隠れ層の
ニューロン数: 15
重み W_H : 784x15次元
バイアス b_H : 15次元

出力層の
ニューロン数: 10
重み W_O : 15x10次元
バイアス b_O : 10次元



こっこのグラフの方が、ずっとわかりやすい！





グラフを流れる量 -- テンソル



グラフを流れる量 -- テンソル

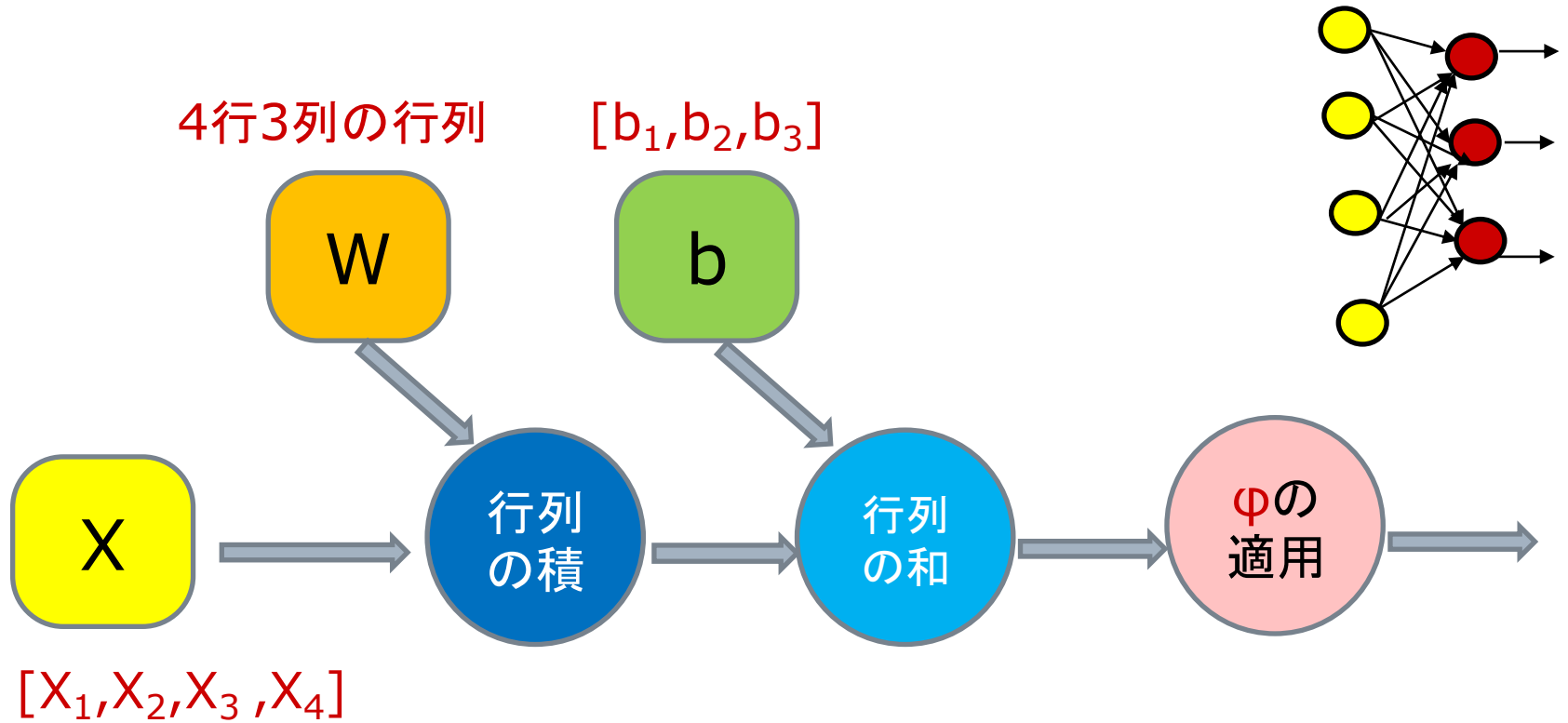
旧来のグラフの沢山の線は、どこへ行ったのか？

新しいグラフでは、グラフを流れるデータは、すべて「テンソル」にまとめられている。

計算グラフのノード間を移動するのは、テンソルだけである。



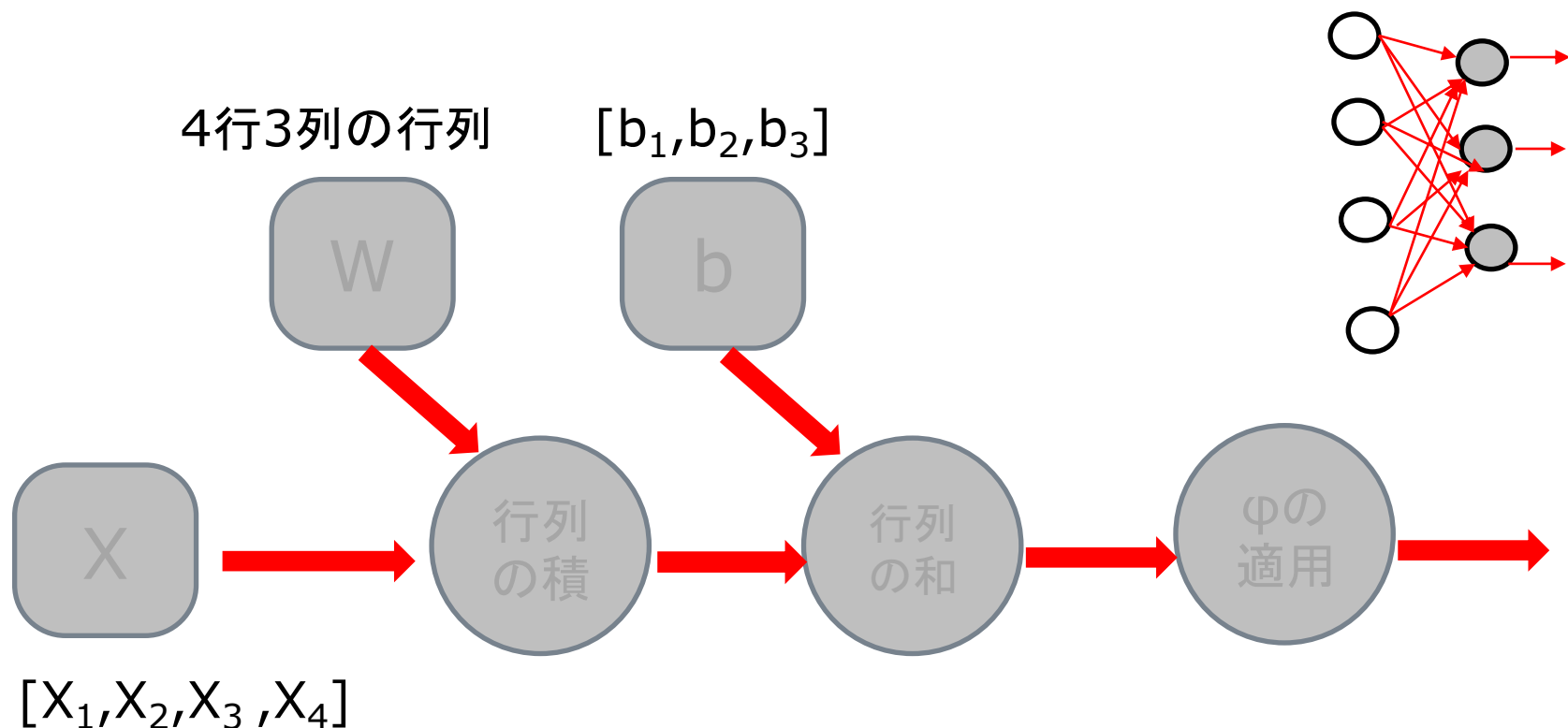
4つの入力を受け取り、3つの出力を返す、3つのニューロンからなる「層」を考えてみよう。



ただ、ここでは、ノードではなく辺の方に注目してみよう。
どのようなタイプのデータが、辺の上を流れるかを考えよう。



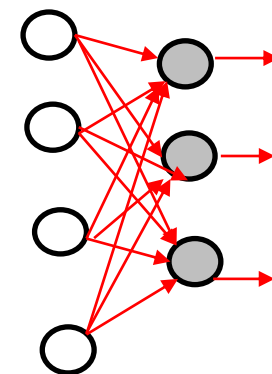
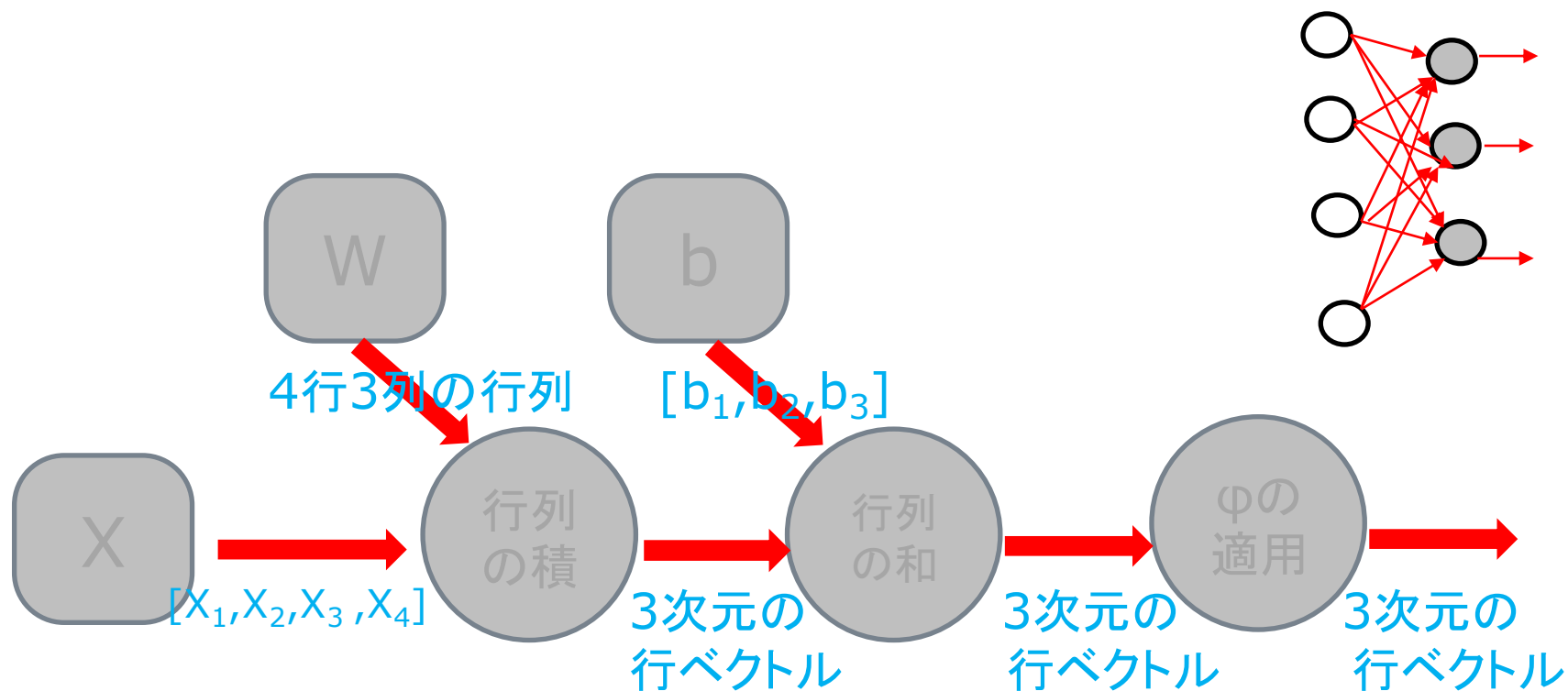
4つの入力を受け取り、3つの出力を返す、3つのニューロンの「層」



こうしたデータをすべて**テンソル**と呼ぶ。
どのような、タイプのデータか見てみよう。



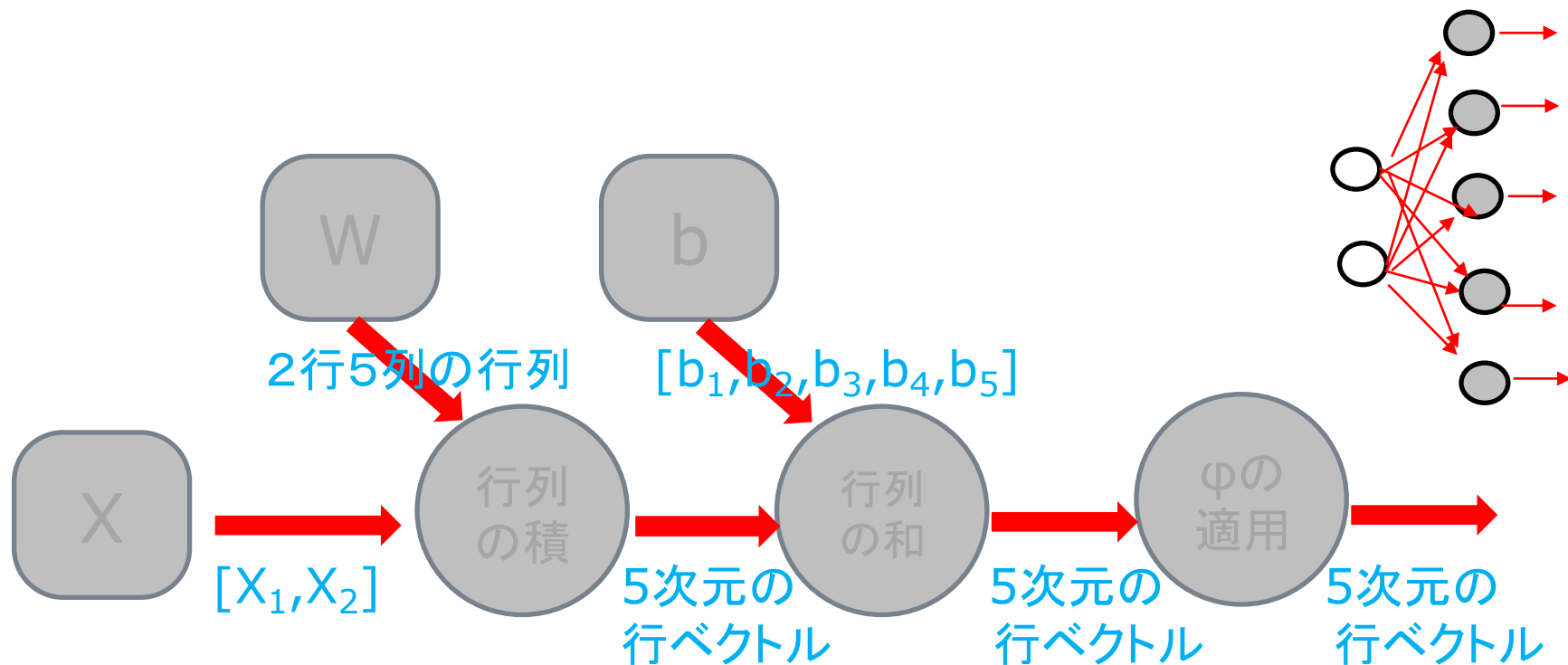
4つの入力を受け取り、3つの出力を返す、3つのニューロンの「層」



テンソルには、いろいろなタイプがある。



2つの入力を受け取り、5つの出力を返す、5つのニューロンの「層」



テンソルには、いろいろなタイプがある。



テンソルのランク(階数)

- テンソルは「ランク」と呼ばれる次元の単位で記述される。
- テンソルのランクは、行列のランクとは異なるものである。
- テンソルのランクは、(時には、位数(order)とか度数(degree)とかn次元とも呼ばれることがあるのだが)、テンソルの次元の数である。
- 例えば、Pythonのリストで定義された次のテンソルのランクは 2である。

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```



テンソルのランク(階数)

- ランク1のテンソルは、「ベクトル」である。
- ランク2のテンソルで、われわれが典型的に考えるのは「行列」である。
- ランク2のテンソルに対しては、 $t[i,j]$ という構文で、任意の要素にアクセスできる。
- ランク3のテンソルには、 $t[i,j,k]$ でアクセスする必要がある。



テンソルのランク(階数)

ランク	数学的実体	Pythonでの表記
0	スカラー(大きさのみ)	<code>s = 483</code>
1	ベクトル(大きさと向き)	<code>v = [1.1, 2.2, 3.3]</code>
2	行列 (数字のテーブル)	<code>m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]</code>
3	3-テンソル	<code>t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]</code>
n	n-Tensor





ニューラル・ネットワークの グラフ表現とモジュール構造

