

2018年度Sセメスター 学術フロンティア講義

深層学習はどのように賢くなるのか

瀧 雅人 (Masato Taki)

RIKEN, iTHEMS

2018.7/4

@東京大学教養学部

?

深層学習はどのように賢くなるのか

?

毎日のようにAI(人工知能)のニュースや話。

今日の記事から見つ
けた某ニュース

AI時代は
営業は足で稼げ！

みたいな本とか
(AI関係ないじゃん！)

いささか食傷気味…

玉石いろんな話がごたまぜに流れてくる感じがしますが、背後には、確かな技術的進展。実態と無関係なhypeは落ち着いても、発展が続くことには間違いない。

様々なアプローチ
(いい意味でなんでもあり)

人工知能
AI

アプローチの一つ

深層学習
Deep Learning

手法群の一つ

機械学習
Machine Learning

人工知能
AI

機運を盛り
上げる

機械学習

Machine Learning

深層学習
Deep Learning

ここで、この6年に
とても大きな発展が！

深層学習＝機械学習の一例

学習や訓練とは？

コンピュータプログラム(機械)がデータや経験を通じていろんな作業を上手くこなせるようにさせること。

今後 も 使い回すため
一部省略:-)

1956 Arthur Samuel's Checker-Playing Program



<https://www.sutori.com/item/1956-arthur-samuel-the-world-s-first-self-learning-program-machine-learning>

初の現代的な機械学習システム

それから60...

2016-2017

ALPHAGO

A DOCUMENTARY • SPRING 2017



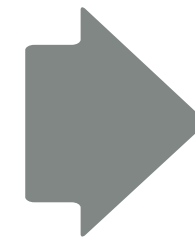
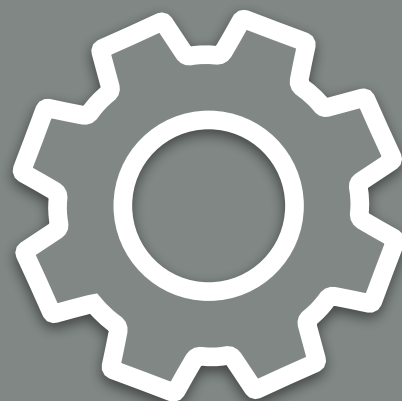
<https://ja-jp.facebook.com/alphagomovie/>

1. 深層学習とは？

データによる機械学習

機械学習 Machine Learning

機械（モデル）



与えられた
課題をこなす

機械学習 Machine Learning

目的に応じて我々が選ぶ。その正体は数式やコンピュータプログラム。



与えられた
課題をこなす

機械学習 Machine Learning

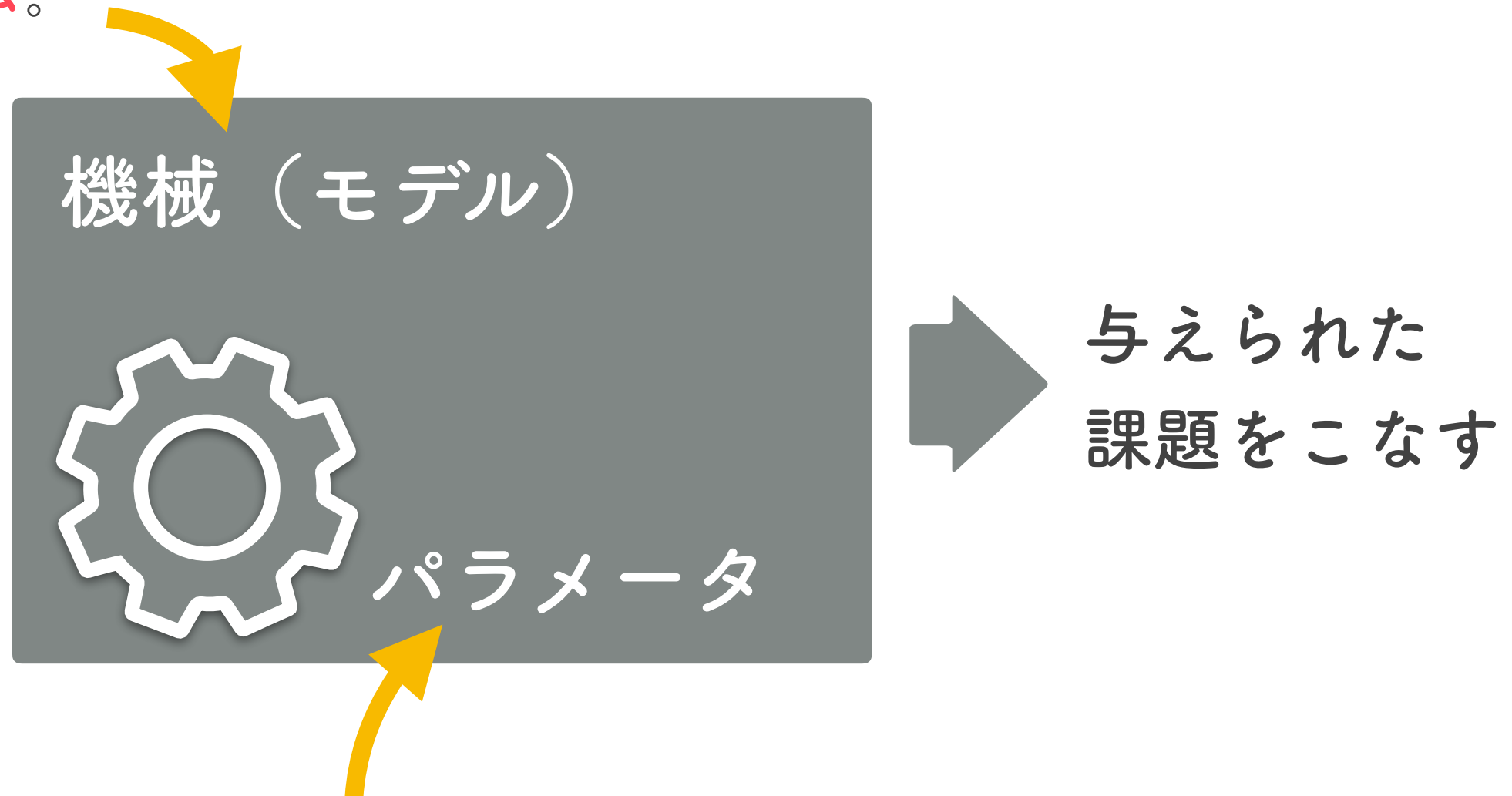
目的に応じて我々が選ぶ。その正体は数式やコンピュータプログラム。



与えられた
課題をこなす

機械学習 Machine Learning

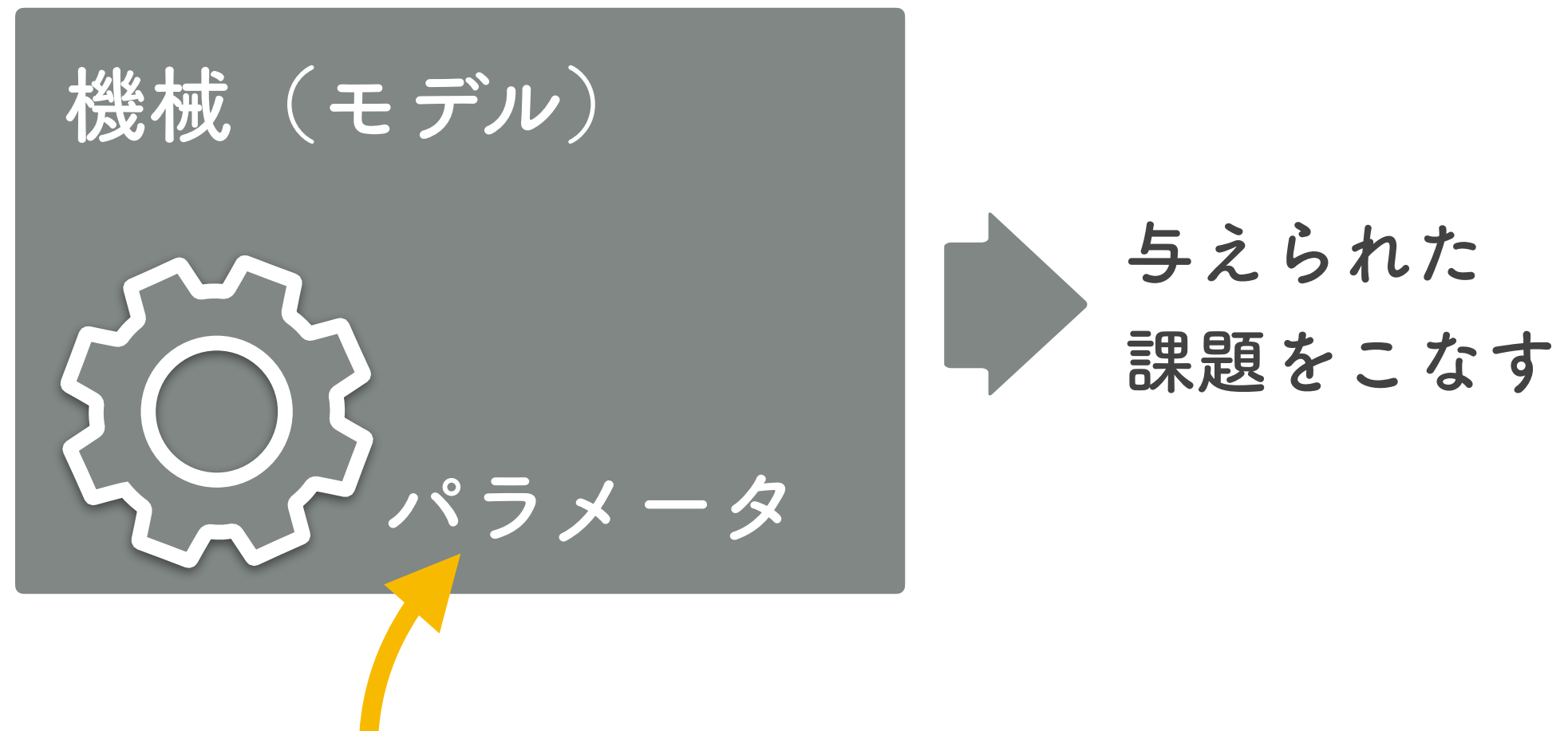
目的に応じて我々が選ぶ。その正体は数式やコンピュータプログラム。



モデルの動作を決める部分。この「つまみ」を、データを使って調節して、課題をうまくこなせるように訓練する(学習させる)。

機械学習 Machine Learning

なぜ我々の手ではなく、データでつまみを回すのか？



モデルの動作を決める部分。この「つまみ」を、データを使って調節して、課題をうまくこなせるように**訓練**する(**学習**させる)。

溢れるデジタルデータ

データさまざま

私ではない仮想的な人の1日

- 7:00** 起床とともに天気をチェック
- 7:05** SNSをチェック
- 7:10** GoogleホームでIoT家電を起動
- 8:00** 近所のスタバでコーヒー購入
- 8:05** パスモで電車に乗車
- 8:30** 研究所到着。オートロック解除
- 12:00** アマゾンでミネラルウォーター購入
- 12:30** 午後から行く駒場への路線検索
- 12:40** クレジットで来週の出張の新幹線チケット購入

...

データさまざま

私たちは普通の一日の生活の中でも、知らぬ間に大量のデータを生み出している。

データさまざま

私たちは普通の一日の生活の中でも、知らぬ間に大量のデータを生み出している。



1年半ごとにデジタルソーシャルデータの量は倍増

多くの産業が、今やこのようなデータに多くを負っている

データさまざま

素粒子加速器実験

衛星による天文観測

などなど、科学の現場も今莫大なデータで溢れかえっている。
全てには手が回らない…

データさまざま

「データは21世紀の石油」 A.ワイガンド『アマゾノミクス』

データさまざま

「データは21世紀の石油」 A.ワイガンド『アマゾノミクス』

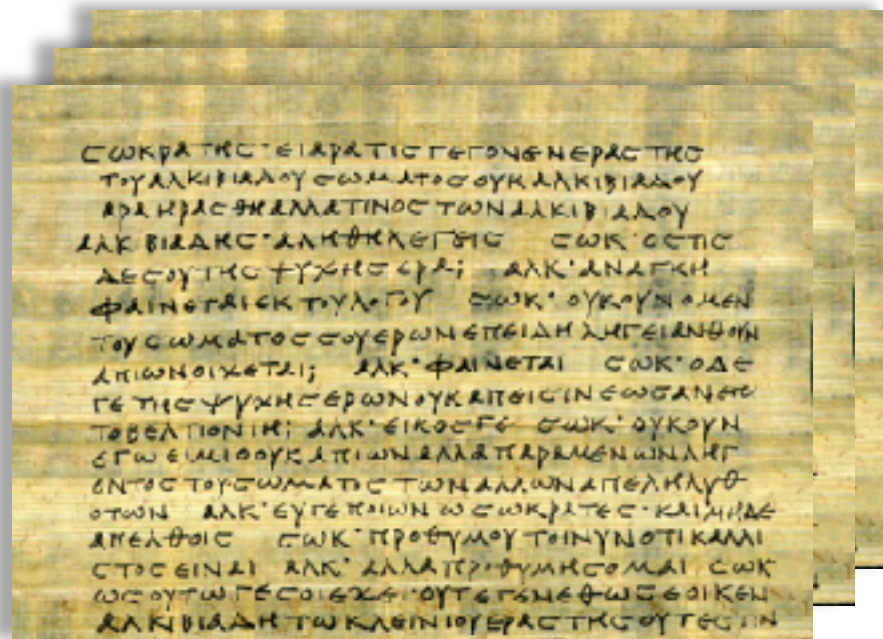
というわけで石油王になりましょう。

データさまざま

データ

基本的に全てのデータは数値の形をとる

(コンピュータや計測機器の上でデジタルに収集・処理されている)



→ デジタル写真 → 文字

→ 数値表現

タスク(task、課題)さまざま

データでどんな課題を学習させるか？

タスク(task、課題)さまざま

手法で分類

回帰

分類

クラスタリング

次元削減

生成モデル

...

目的で分類

予測

機械翻訳

異常検知

データ生成

ゲームAI

...

...

タスク(task、課題)さまざま

機械学習で自動化されるためには、コンピュータ上でタスクが表現できなければならない

————→ 基本的にタスクも数値の形をとる

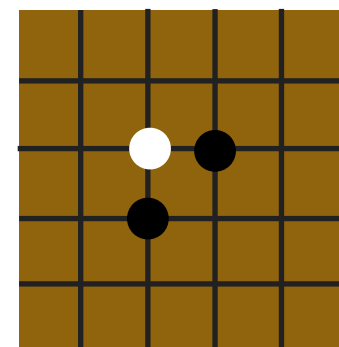
タスク(task、課題)さまざま

機械学習で自動化されるためには、コンピュータ上でタスクが表現できなければならない

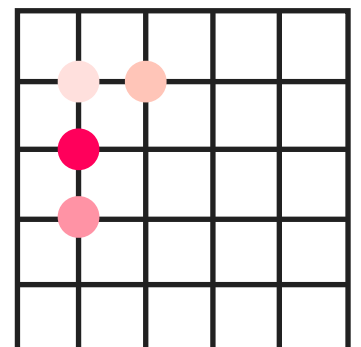
→ 基本的にタスクも数値の形をとる

eg. AlphaGo

入力：現在の碁の盤面



出力：自分が石を置くべき盤面の位置
(など)



機械学習 Machine Learning

結局、機械学習は多くの場合、

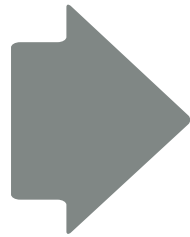


機械学習 Machine Learning

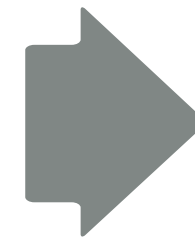
画像認識



https://ja.wikipedia.org/wiki/ネコ#/media/File:Cat_Eyes.jpg



機械（モデル）

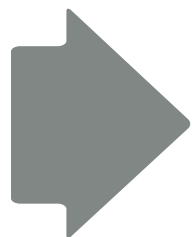


Cat

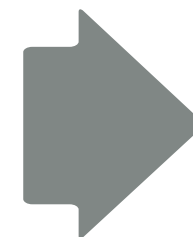
機械学習 Machine Learning

機械翻訳

这是一支钢笔



機械（モデル）



This is a pen

機械学習 Machine Learning

つまり与えられた数値(入力)から、予測数値(出力)を吐き出す
「予言ボックス」を作ればいい

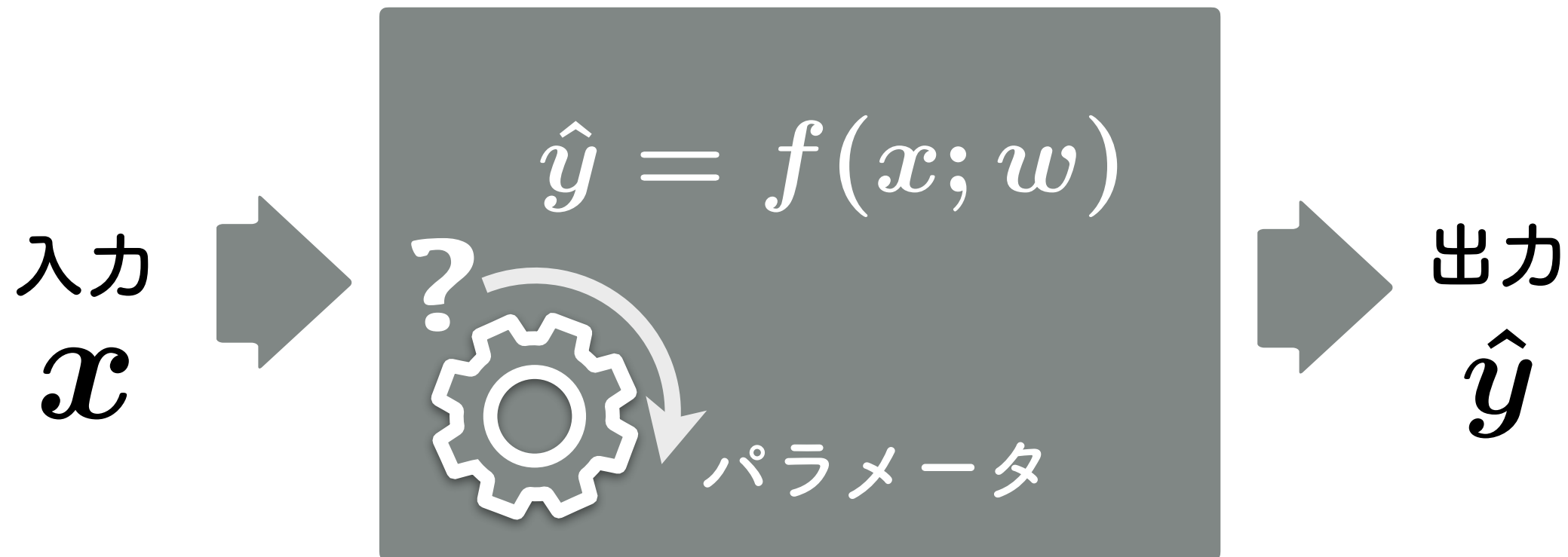


$$y = f(x)$$

どんな関数が答えかは、初めは見当もつかない

機械学習 Machine Learning

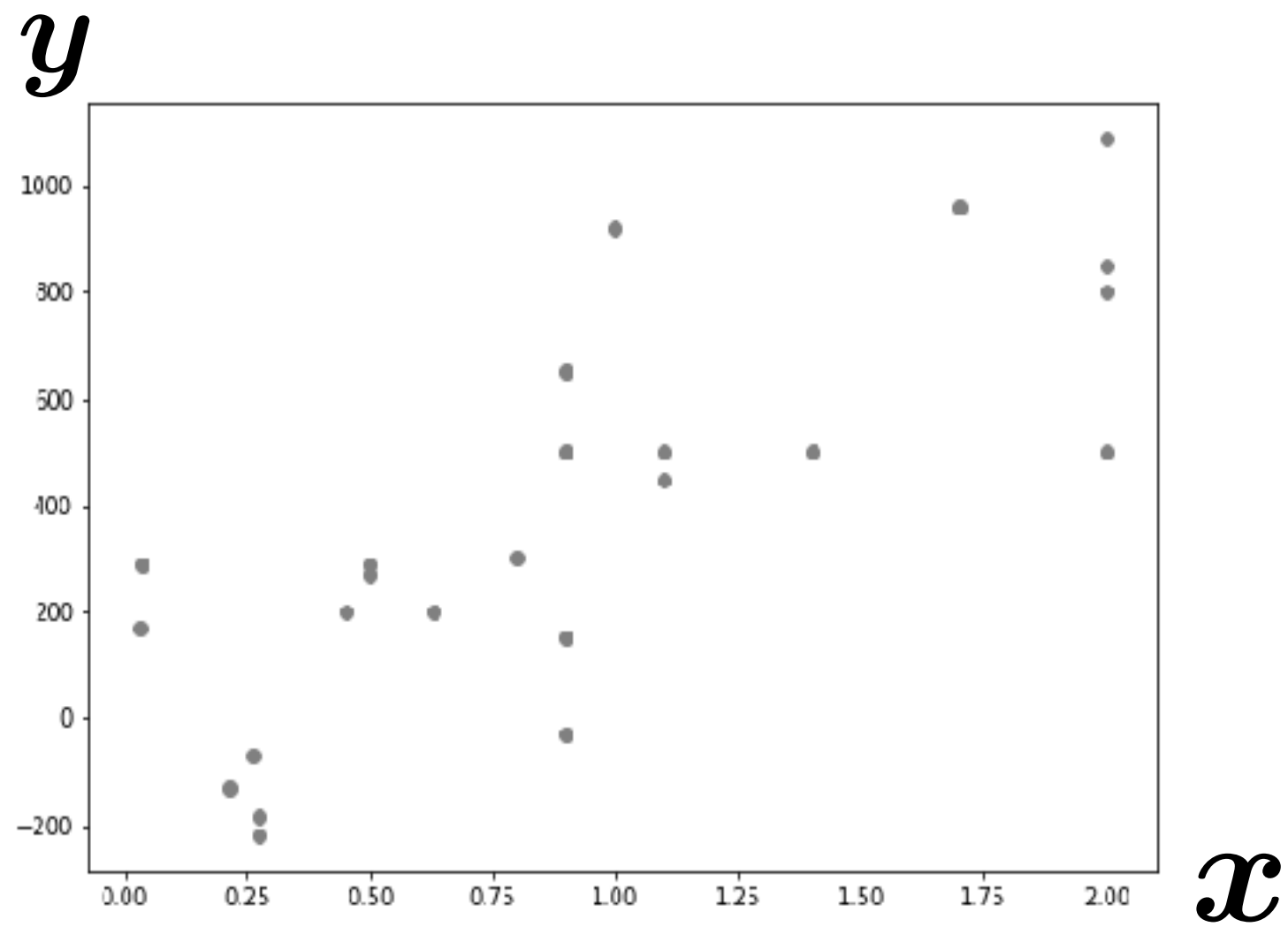
つまり与えられた数値(入力)から、予測数値(出力)を吐き出す
「予言ボックス」を作ればいい



データに一番よくマッチするパラメータ w を探す問題に！
(問題が扱いやすくなった)

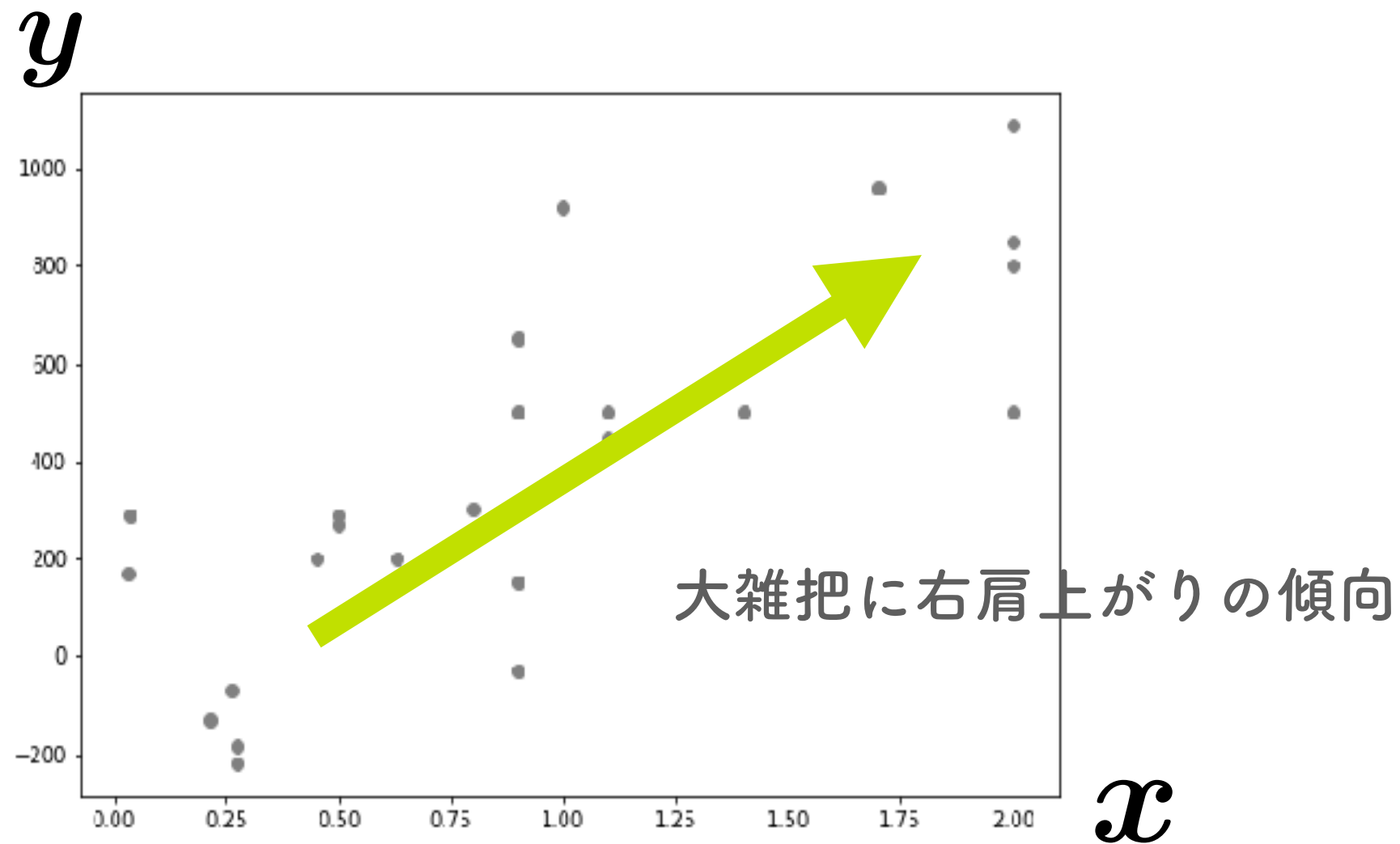
一番簡単で有名なマシン
について説明します
(単回帰)

回帰マシン



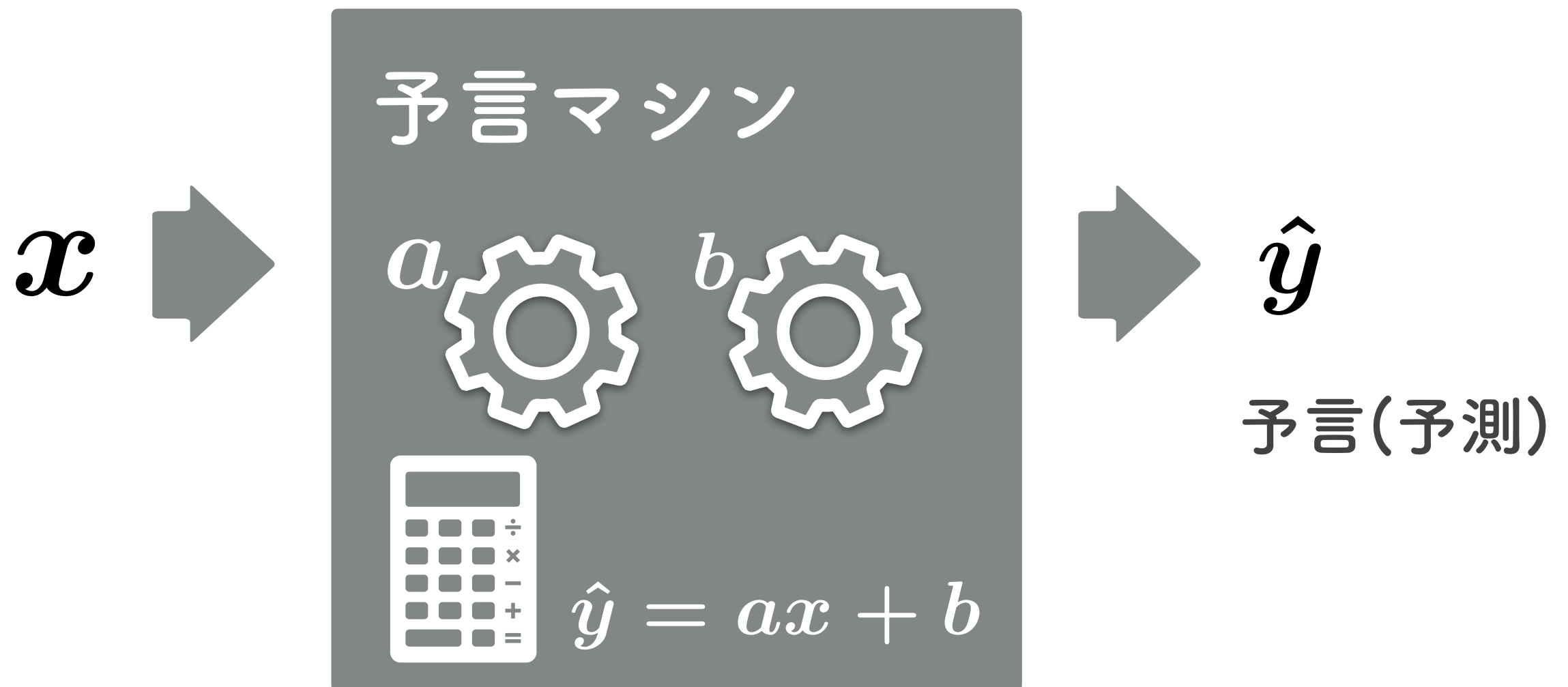
とあるデータ

回帰マシン

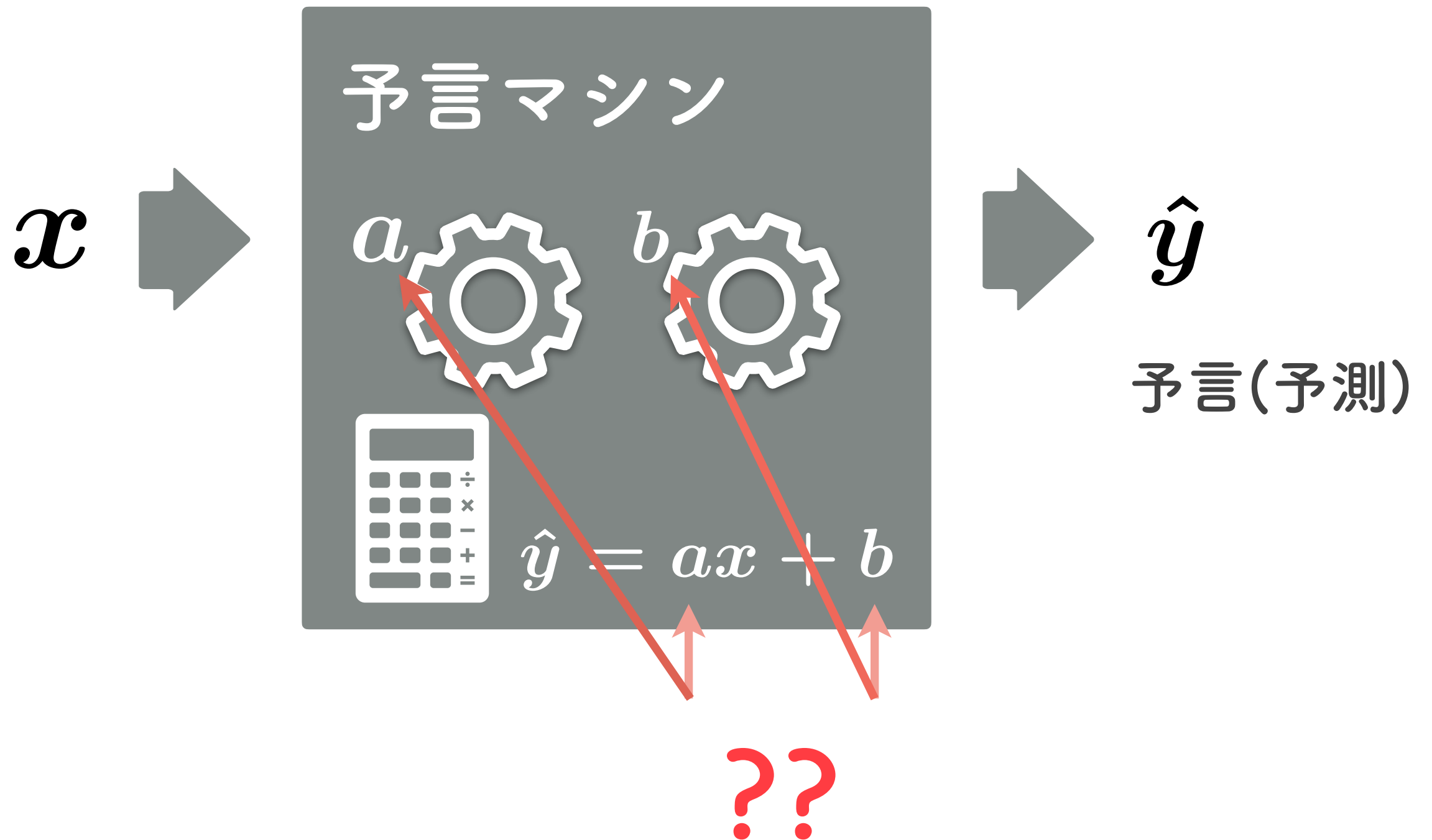


モデル $\hat{y} = ax + b$

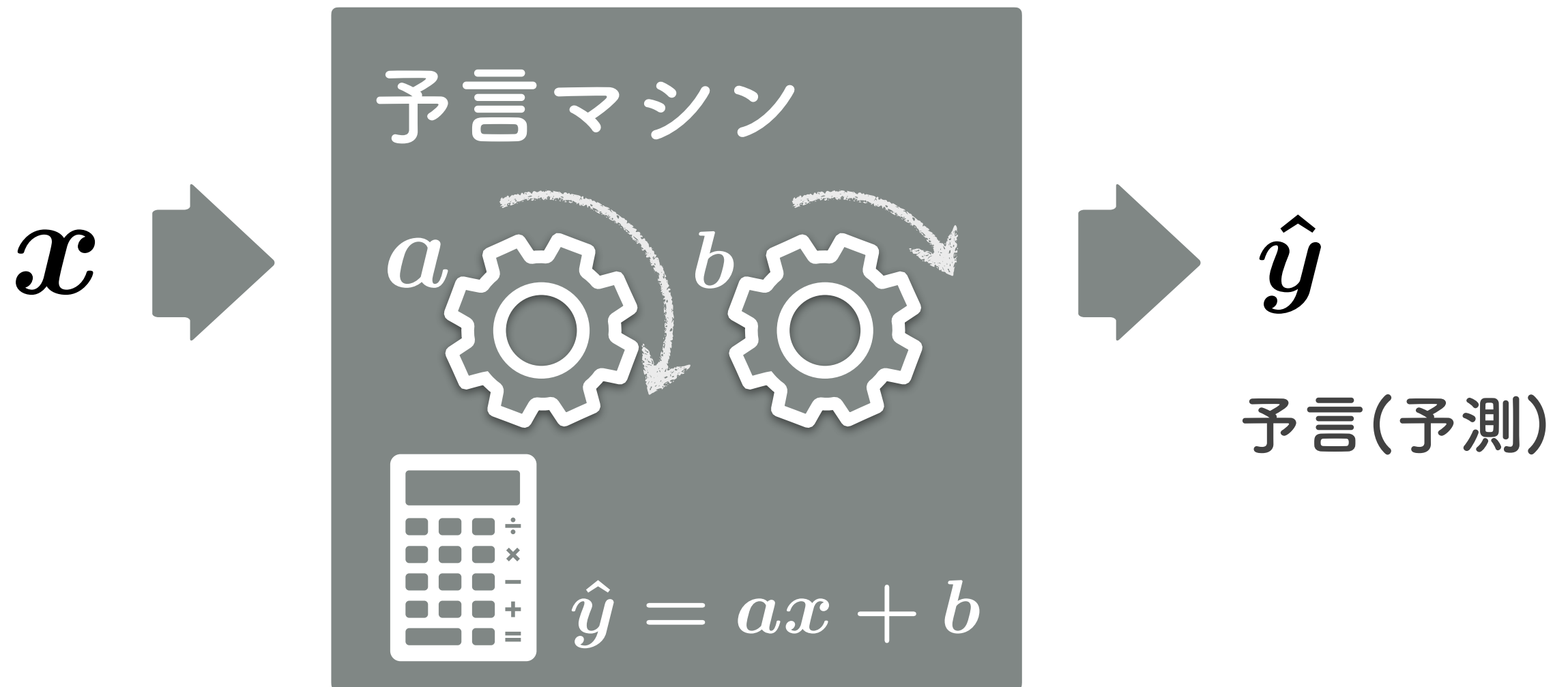
回帰マシン



回帰マシン



回帰マシン



学習 (learning) : データから、最もうまく予言
できるパラメータを決める

回帰マシン

データの学習

x	y
0.032	170
0.034	290
0.214	-130
0.263	-70
...	
2.000	1090

回帰マシン

データの学習



x	y
0.032	170
0.034	290
0.214	-130
0.263	-70
...	
2.000	1090

一個のデータ点
(インスタンス、事例)

回帰マシン

データの学習

この一つのデータ点に関する、予測のズレ

x	y
0.214	-130

$$\hat{y} - y = (a \times 0.214 + b) - (-130)$$

マシンの予言 正解の値

回帰マシン

データの学習

この一つのデータ点に関する、予測のズレ

x	y
0.214	-130

$$\begin{aligned}\hat{y} - y &= (a \times 0.214 + b) - (-130) \\ &= \hat{y}(\underset{x\text{の値}}{0.214}; a, b) - y\end{aligned}$$

回帰マシン

データの学習

各データ点でのズレの自乗を予測の誤差と定義する

$$(\hat{y}(0.214; a, b) - y)^2$$

回帰マシン

データの学習

各データ点でのズレの自乗を予測の誤差と定義する

$$(\hat{y}(0.214; a, b) - y)^2$$

それをすべてのデータ点で平均したものを誤差関数(損失関数)と呼びます

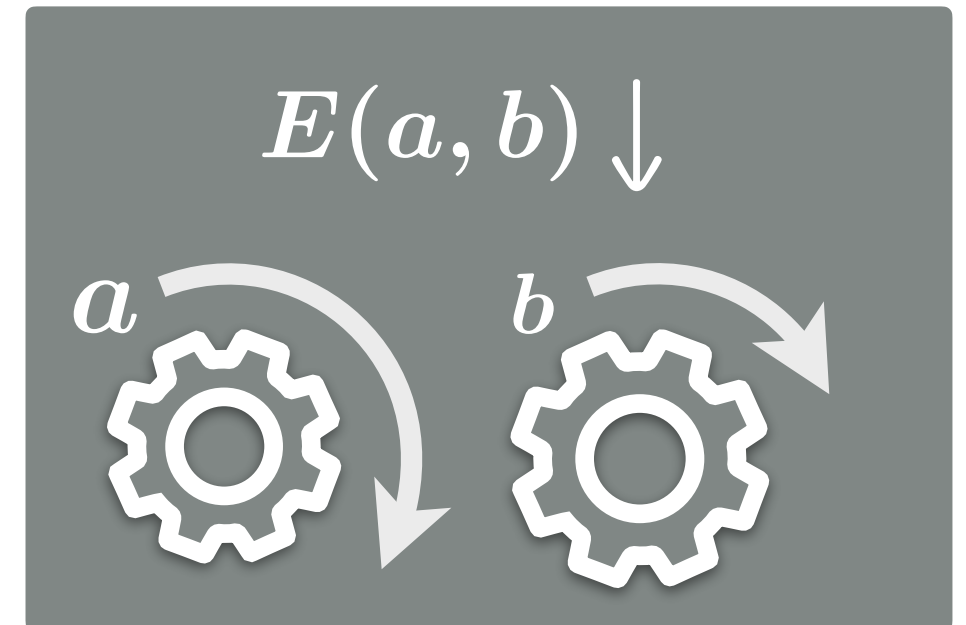
$$E(a, b) = \frac{1}{N} \sum_{n=1}^N (\hat{y}(x_n; a, b) - y_n)^2$$

回帰マシン

データの学習

誤差関数が小さくなるような方向にパラメータのダイヤルを回して調整

すると、どのデータに対しても満遍なく予測誤差が小さいマシンが実現される（だろう）



$$E(a, b) = \frac{1}{N} \sum_{n=1}^N (\hat{y}(x_n; a, b) - y_n)^2$$

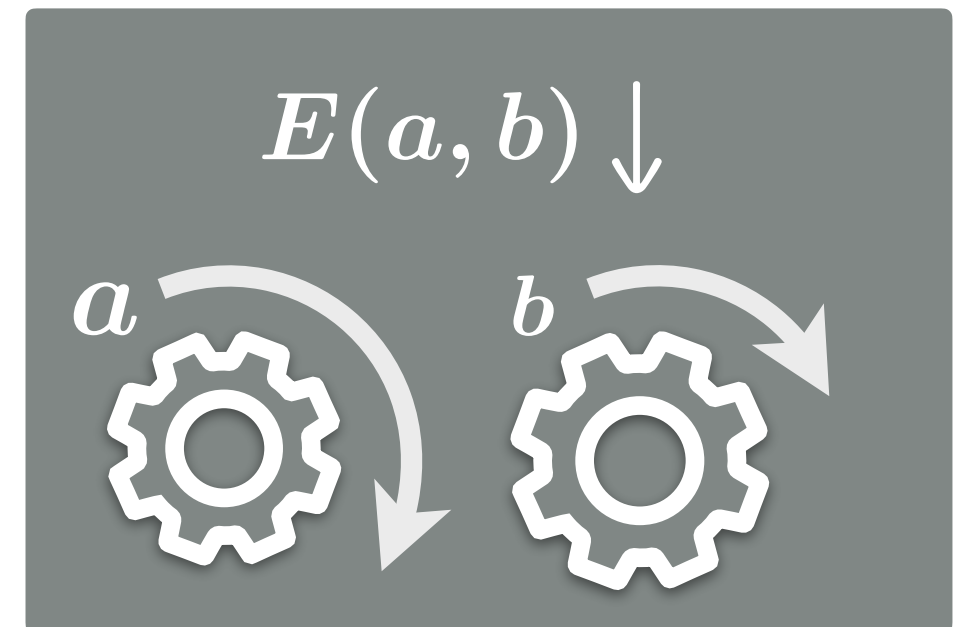
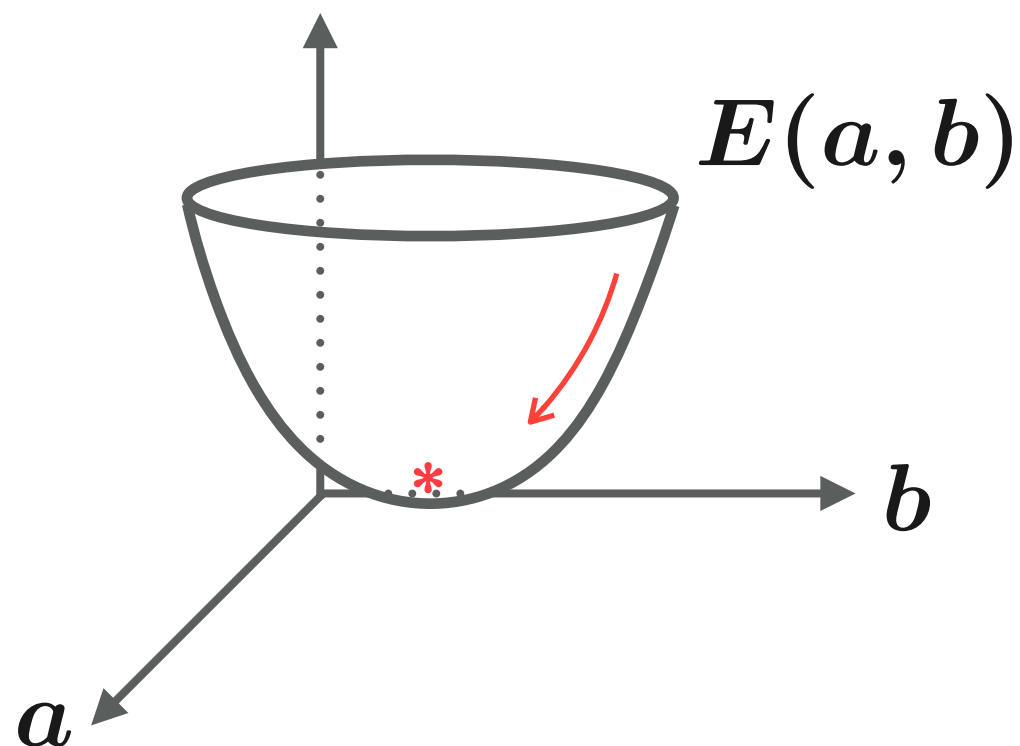
回帰マシン

データの**学習**

||

最適化(最小化)問題

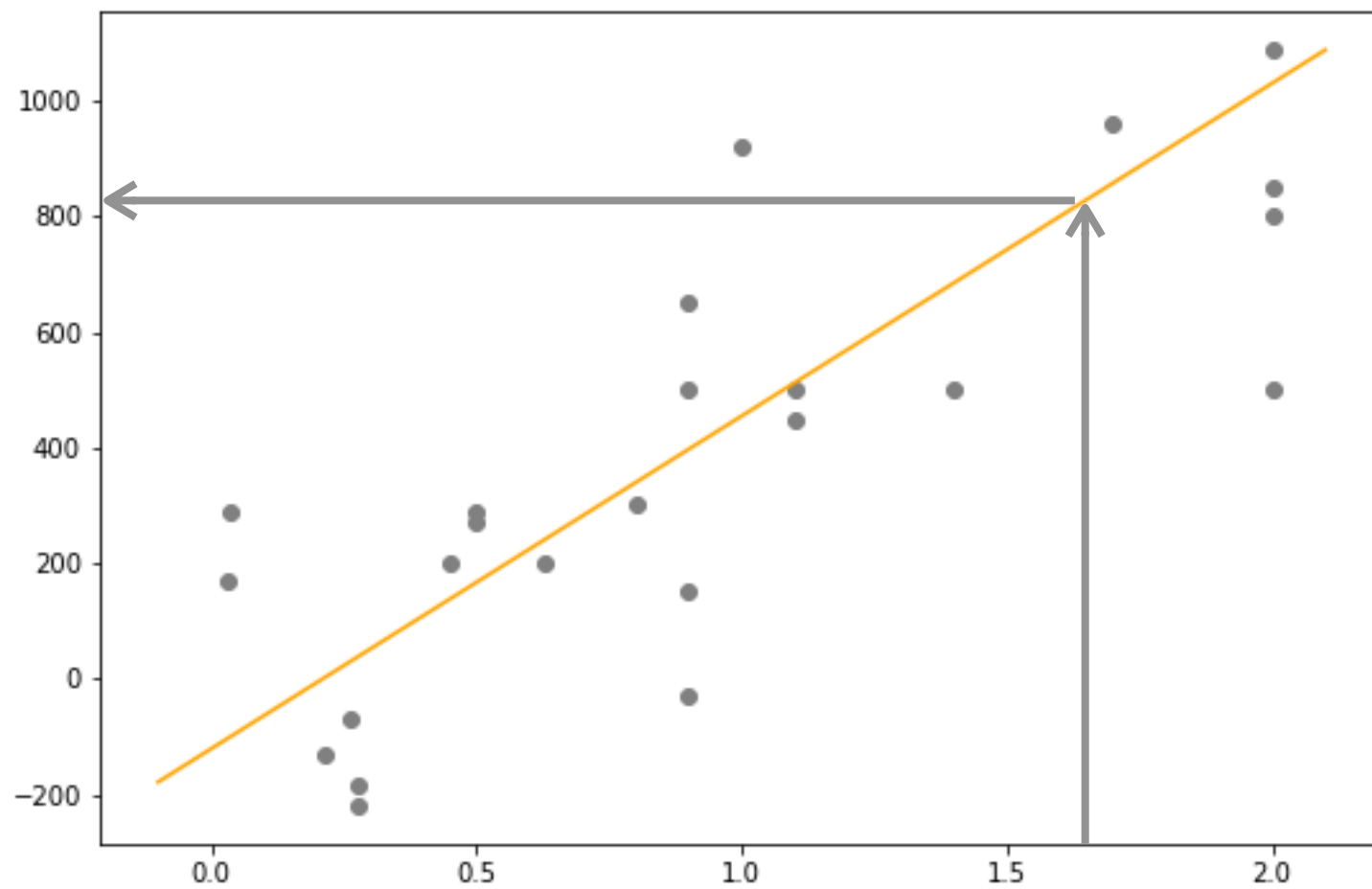
$$a^*, b^* = \operatorname{argmin} E(a, b)$$



これが解けたとすると…

回帰マシン

y



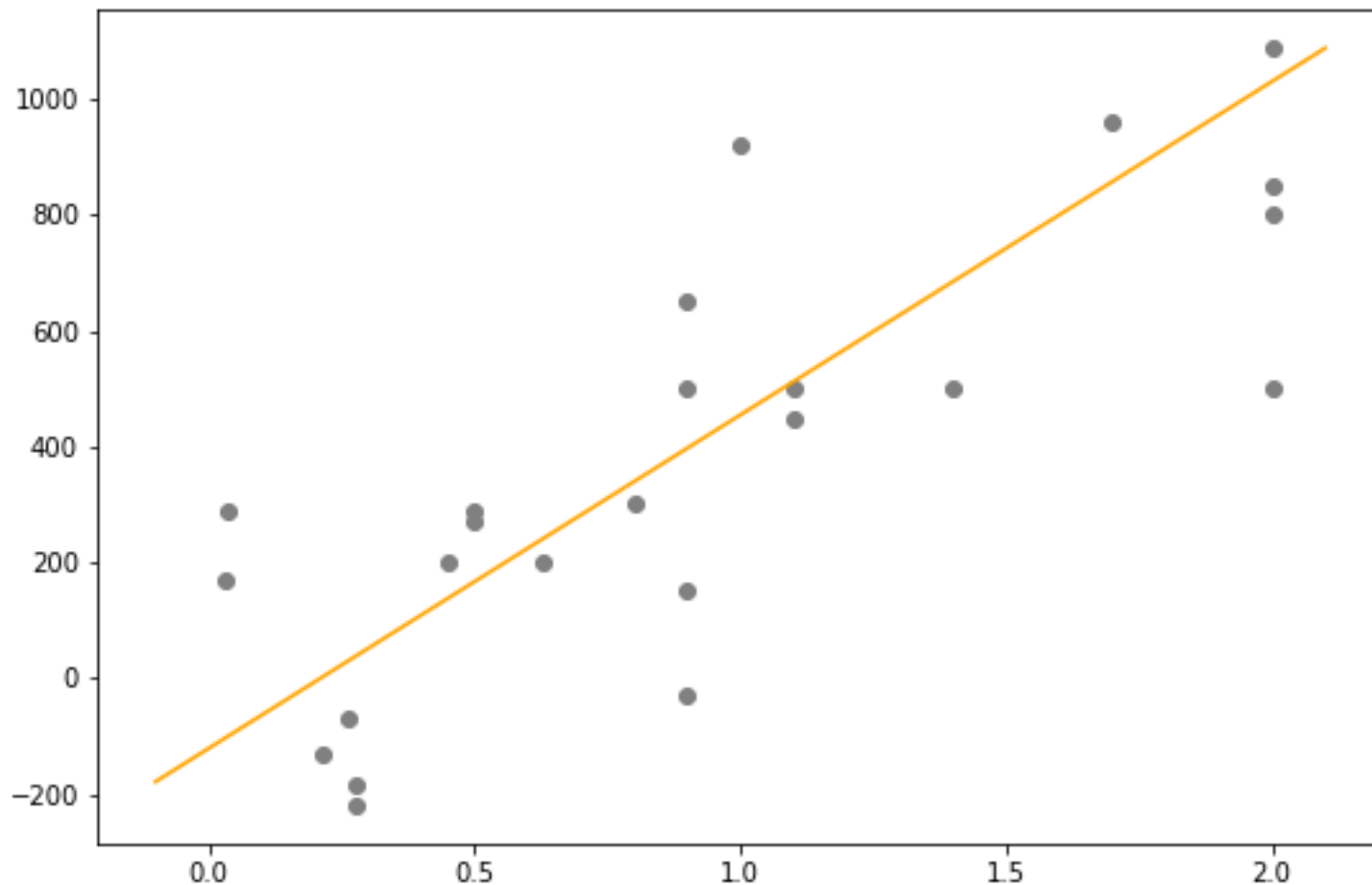
$y =$

$$\hat{y}(x) = a^*x + b^*$$

x

回帰マシン

y



$y =$

$$\hat{y}(x) = a^*x + b^*$$

x = 銀河までの距離 Mpc

y = 銀河の速度 km/s

x

井上さんの授業(4/18)でてきた **ハッブルの法則**！

= 膨張宇宙の証拠

(ハッブル自身のこのデータは、実はダメダメでしたが…)

回帰マシン

銀河までの距離 Mpc

銀河の速度 km/s

x

y

両者に関連(相関)

膨張宇宙

データとモデルで、この関連を捉えられる

(教師あり)学習の考え方 のまとめ

① データを用意する

x = 入力 (input)

y = 出力 (output)

② y の予測問題を解くのに、マシン (=モデル)を用意する

$$\hat{y}(x; a, b)$$

③ データから誤差関数を計算し、最小 化問題をといて、最適なパラメータ a, b を決定する

実際には…

- ① データは多次元、問題は非線形
- ② マシン(=モデル)は多くのパラメータを持つ複雑な関数
- ③ 誤差関数も場合に応じて様々なものを使う。最小化問題をとくのは、一般にそう簡単ではない。などなど

さらに単に最小化するだけではなくて、問題を修正しないと予測性能が出ない(最適化アルゴリズムの工夫、正則化など)

② マシン(=モデル)は多くのパラメータを持つ複雑な関数

まずは一般に、与えられたデータと予測しなくてはならない問題に向けたモデルを用意するにはどうすれば…(→これまでに開発された様々な機械学習から探す)

できることならば高い汎用性と性能を持つ魔法のようなモデルが欲しい。そんなうまい話はあるのか…

深層学習

II

ニューラルネット
を使った機械学習

深層学習(Deep Learning)



このモデルをどう選んだら高い性能が出る??

深層学習(Deep Learning)



我々人間並みに賢い機械を作るには、人間の脳を参考にすれば良い！？

深層学習(Deep Learning)



深層学習は、モデルを**グラフ**を使って直感的に設計できる便利な手法

この**グラフ**は、我々の脳(神経系)を模倣したもの

脳、 とはいふものの

脳をモデル化！？



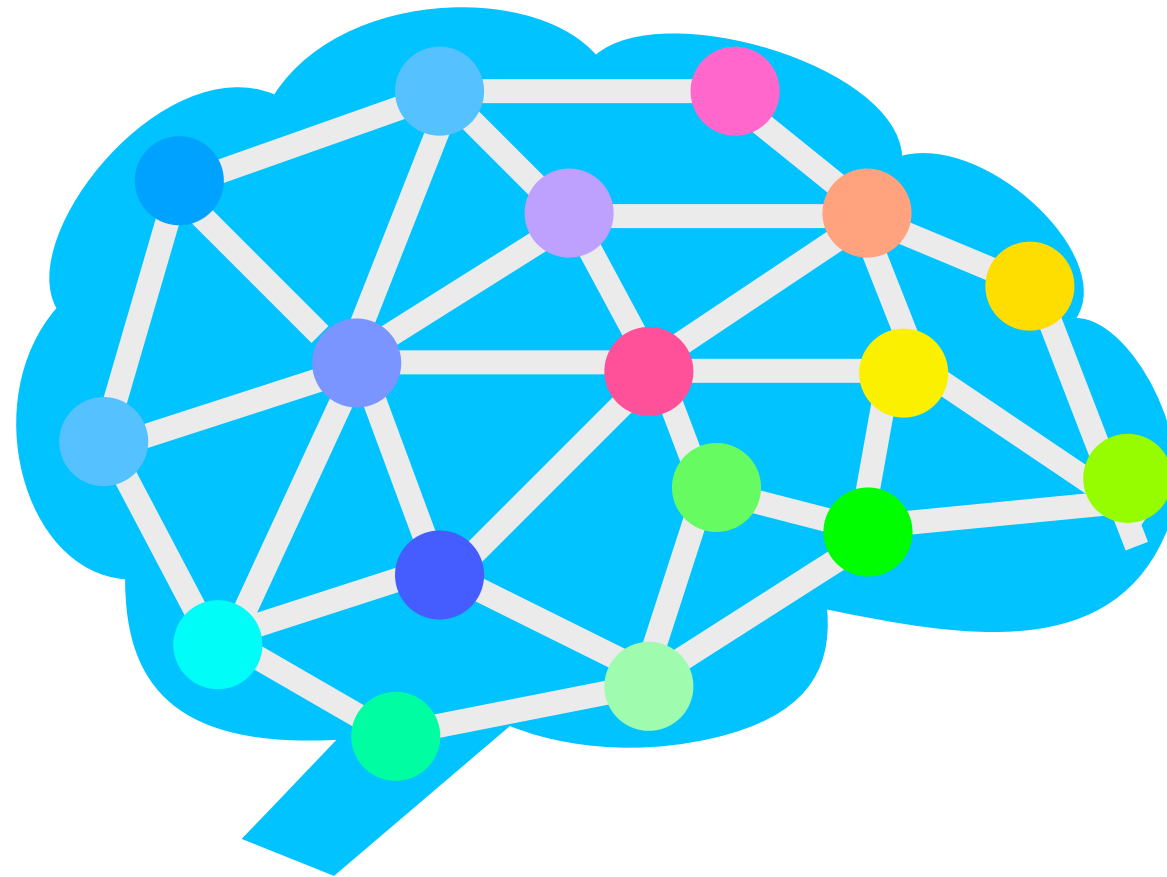
脳をモデル化！？



||

Network

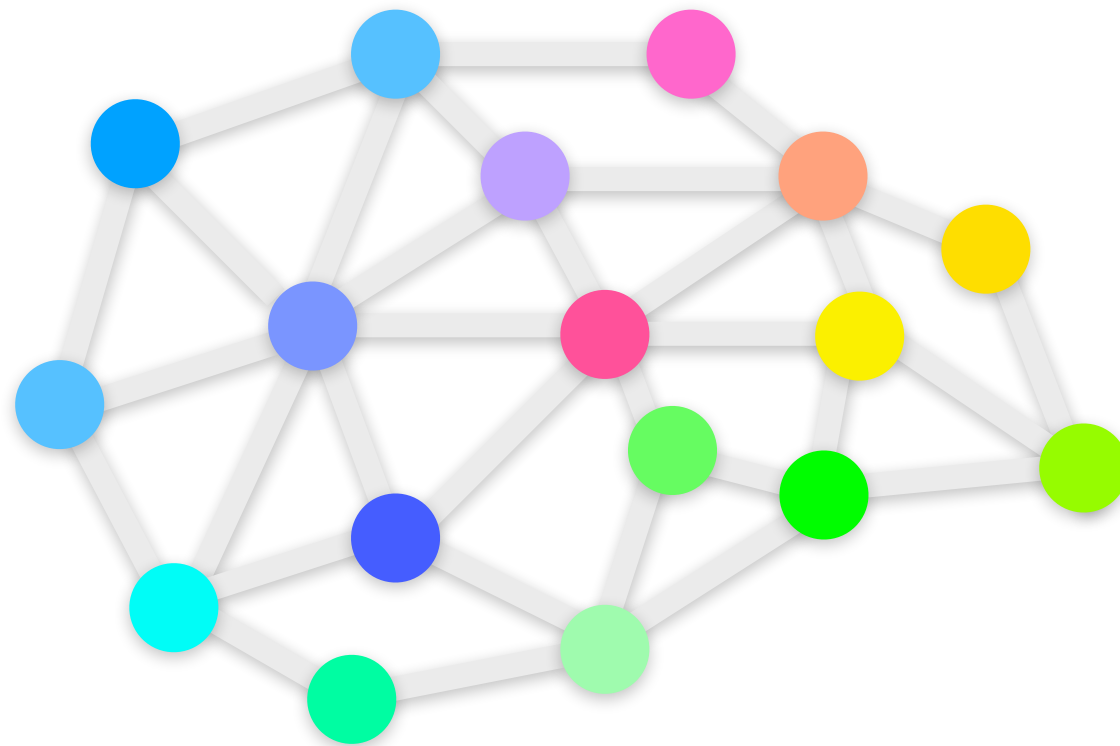
脳をモデル化！？



||

Network

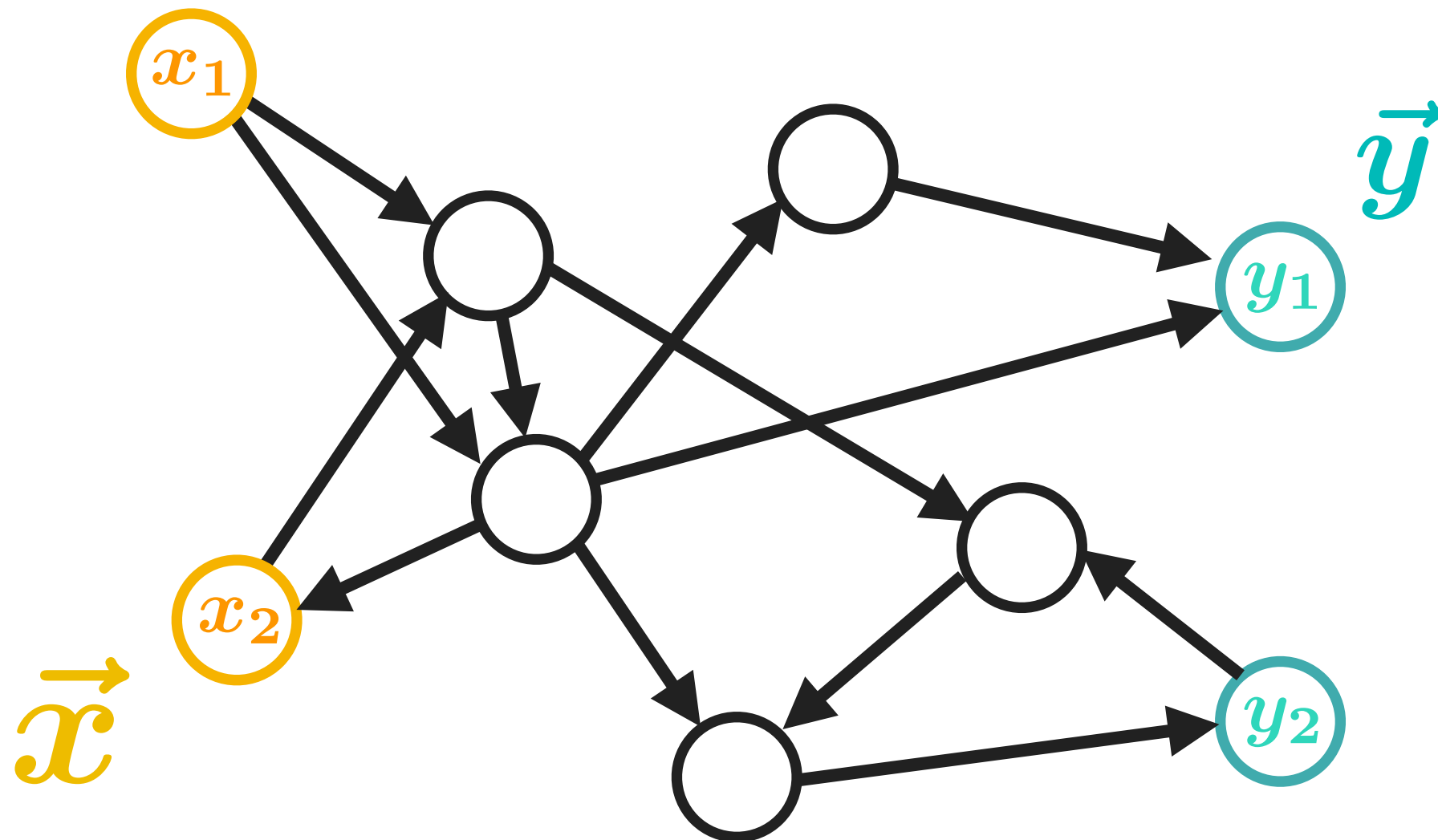
脳をモデル化！？



(Artificial) Neural Network

Neural Network / Deep Learning

モデル = 有向グラフ



Neural Network / Deep Learning

モデル = 有向グラフ

「良いグラフ」を設計することで、難しい
問題がよく学習できる

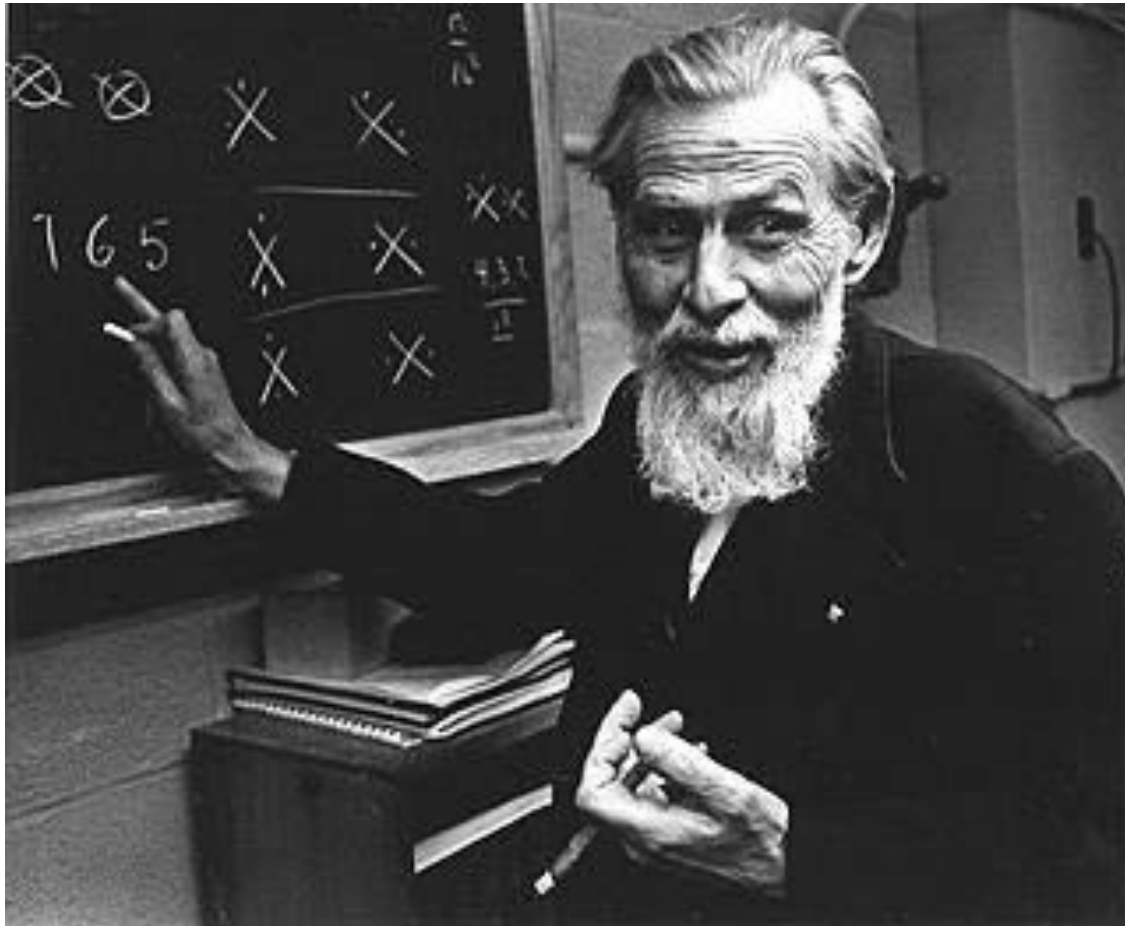
ニューラルネットワーク の起源と基礎

深層学習(ニューラルネット)の起源

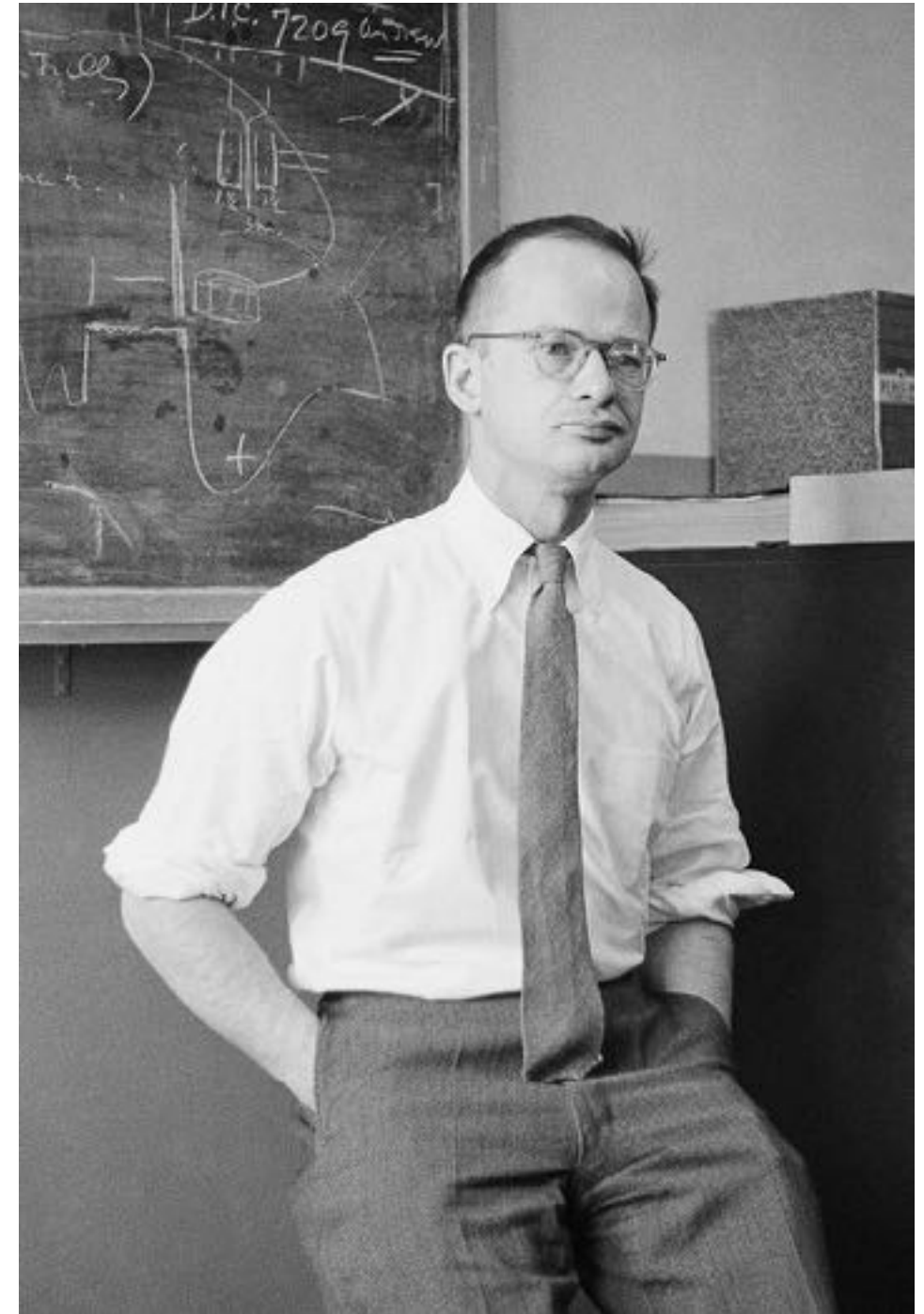
20世紀中盤までさかのぼる

(実はチューリングも人工ニューラルネットについて考察していたが、それは今回は置いておいて)

McCulloch-Pittsの人工ニューロン (1943)

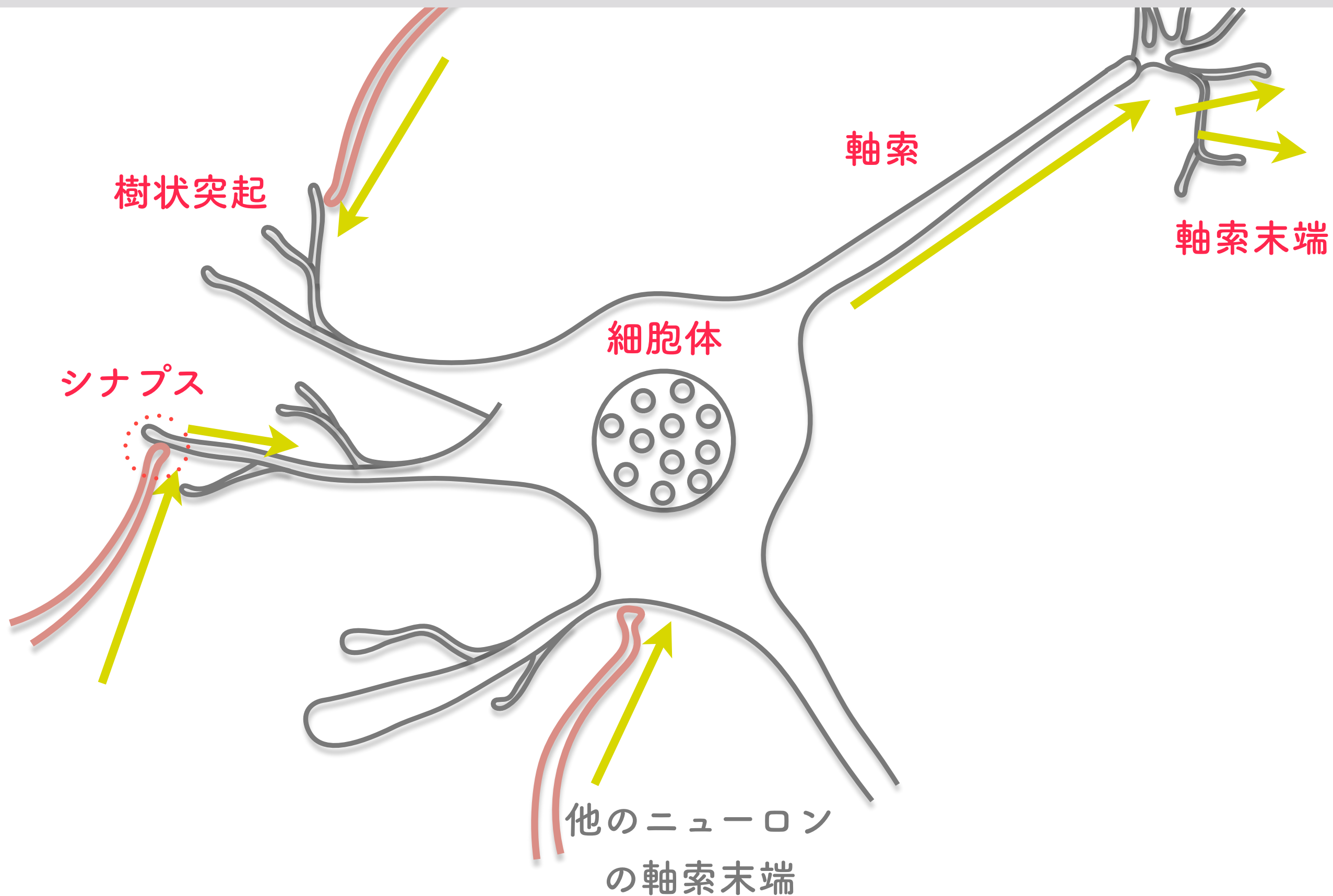


心理学者

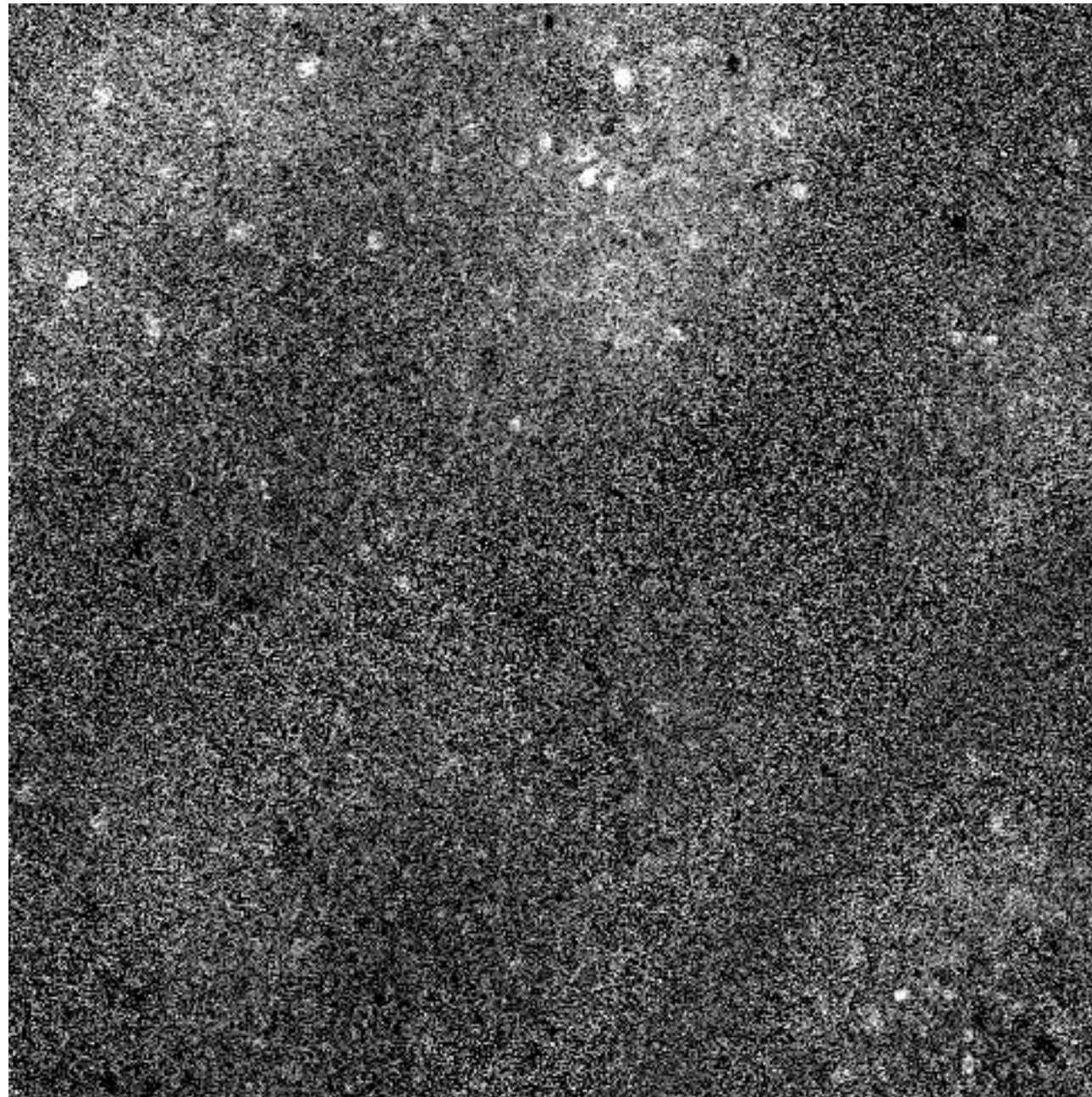


数学者、論理学者

ニューロン（神経細胞）

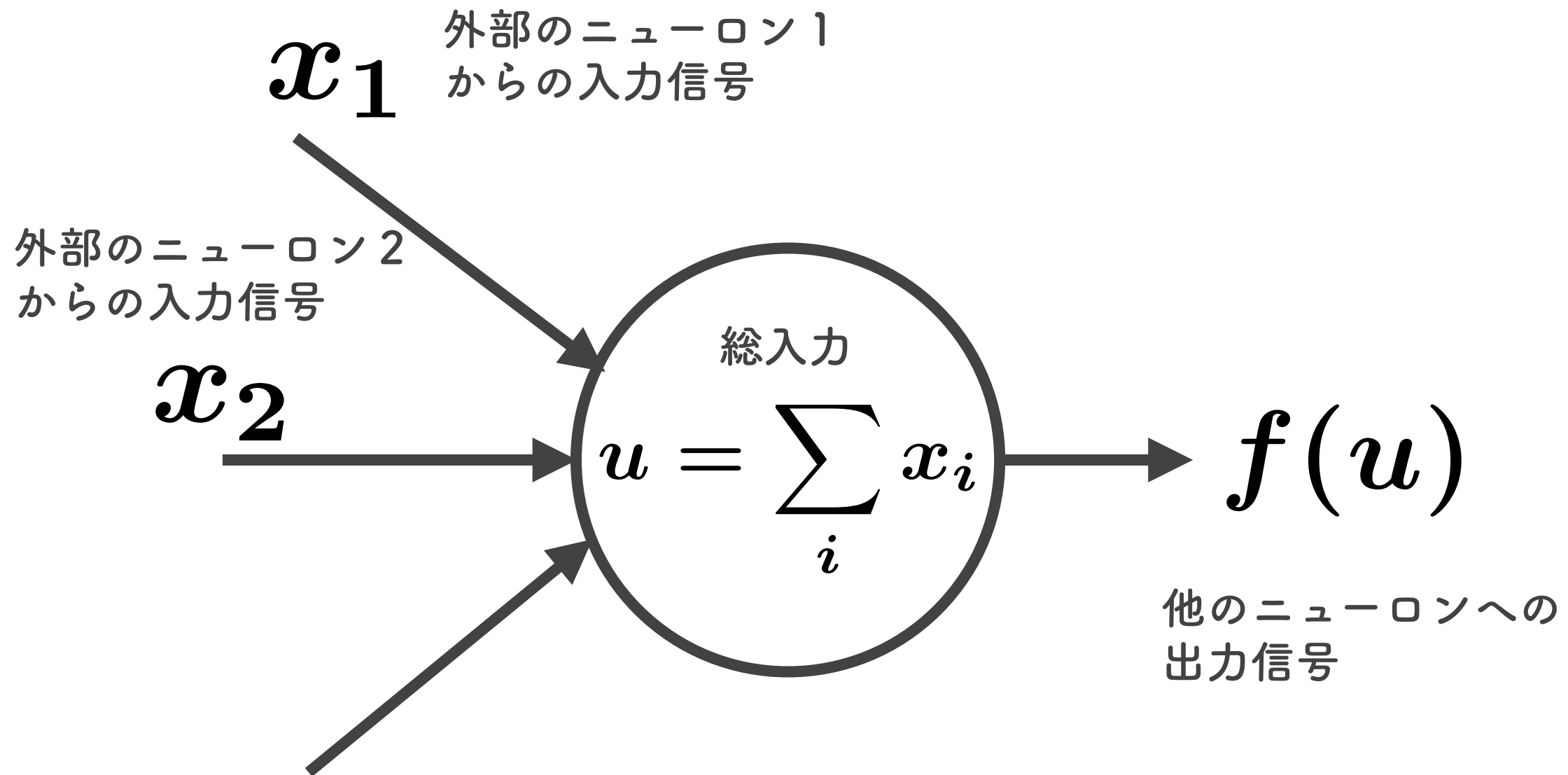


ニューロン（神経細胞）

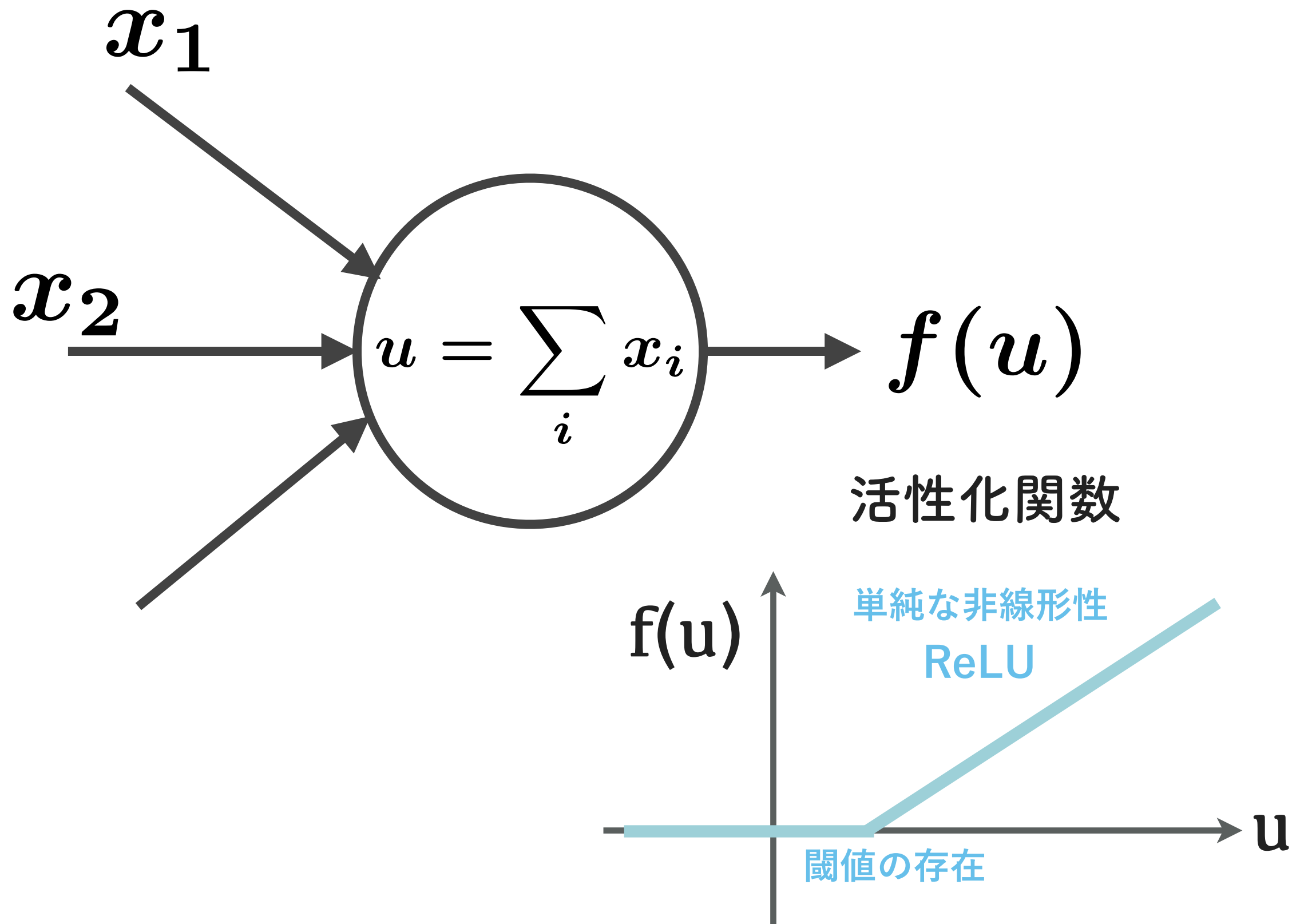


マーモセットの一次視覚野(大脳皮質の深さ250um)
色々な角度の縞模様(視覚刺激)への反応動画
東京大学医学系研究科 松井鉄平先生提供

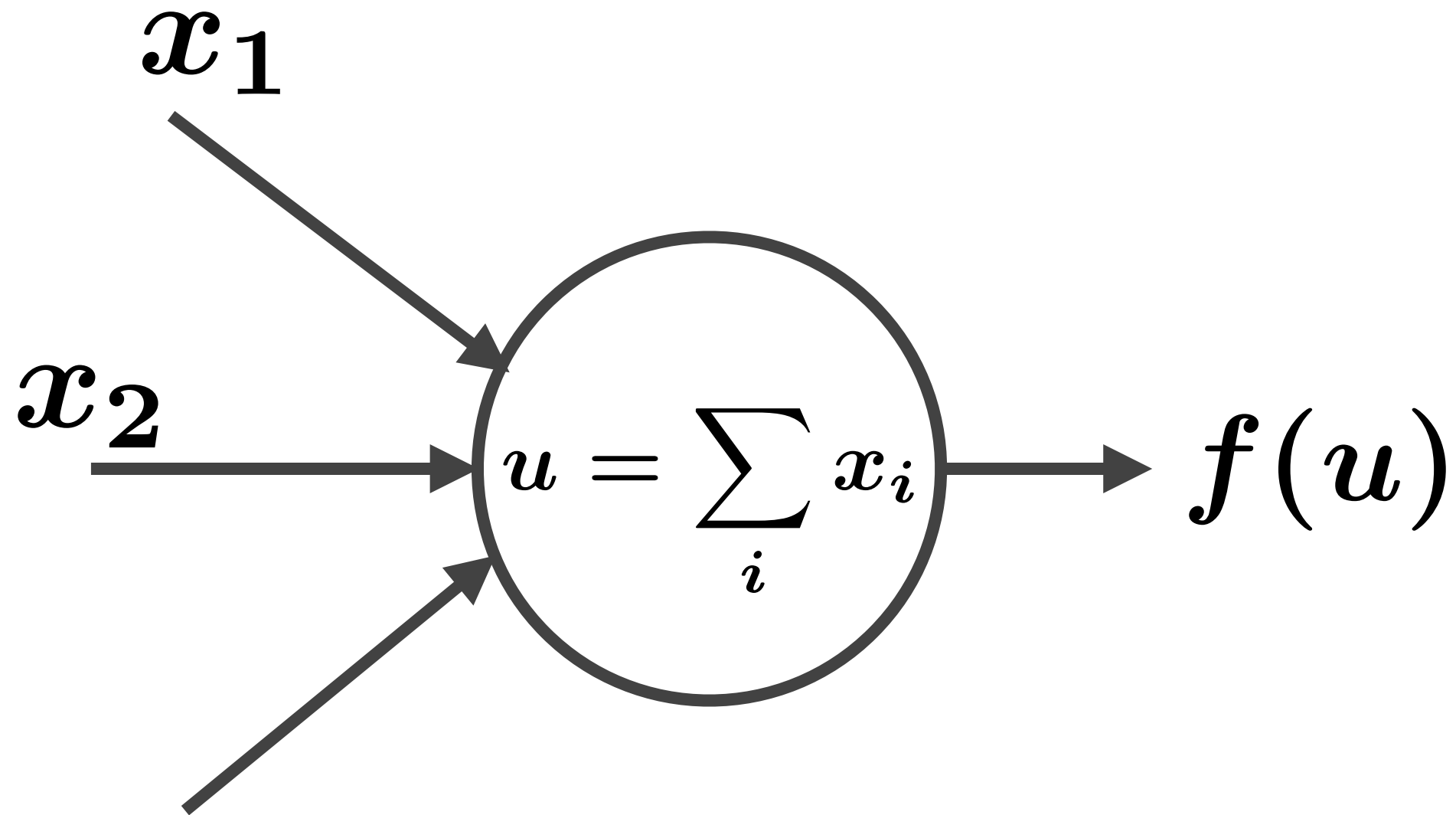
McCulloch-Pittsの人工ニューロン (1943)



McCulloch-Pittsの人工ニューロン (1943)



McCulloch-Pittsの人工ニューロン (1943)



これらの組み合わせで、任意の論理回路(論理計算)が実現できる

McCulloch-Pittsの人工ニューロン (1943)

証明： ニューロンいくつかでNANDゲートを作る

e.g.



$(0, 0) \rightarrow 1$

$(1, 0) \rightarrow 1$

$(0, 1) \rightarrow 1$

$(1, 1) \rightarrow 0$

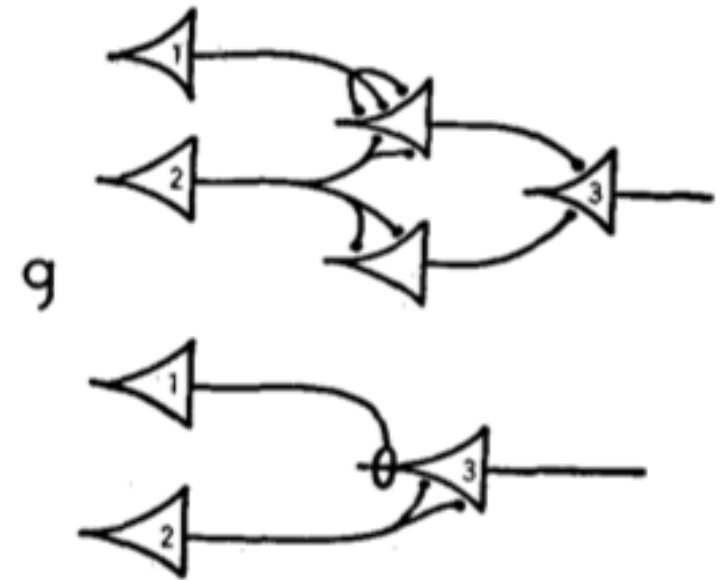
帯、前書き

「NANDからテトリスへ」

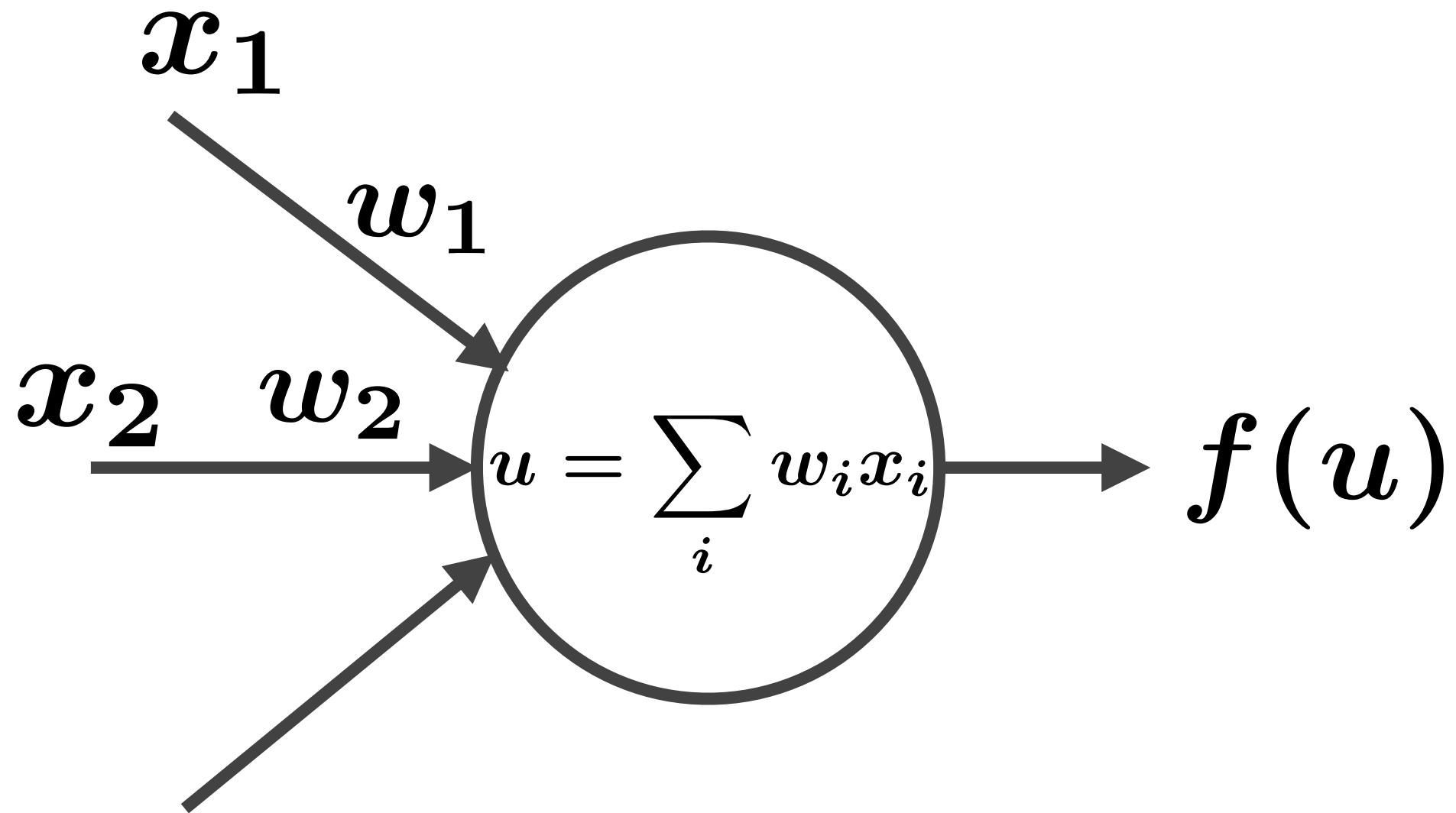
McCulloch-Pittsの人工ニューロン (1943)

どんなコンピュータ(論理回路)も、人工的な神経回路で実現できる

ただ、決まった機能を実現するためには、回路を精密に設計しなくてはならず非実用的



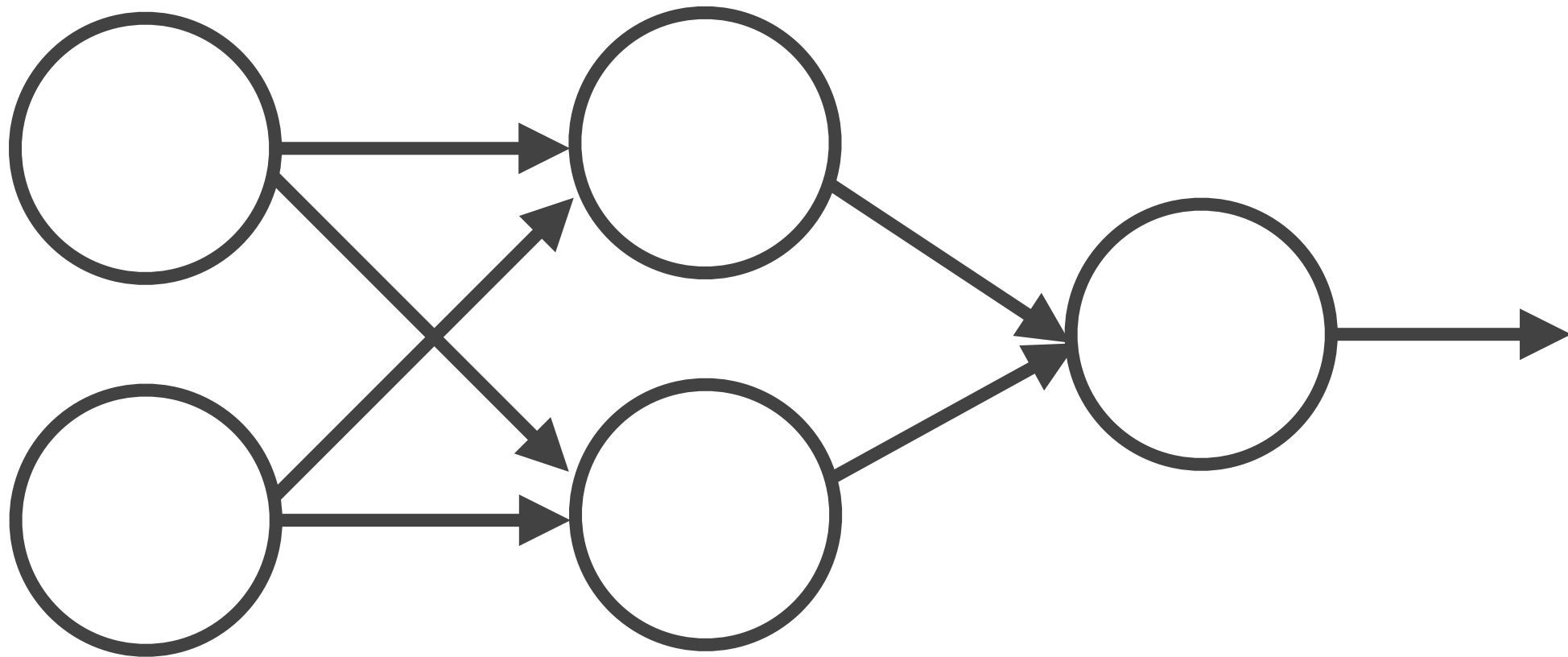
Rosenblattのパーセプトロン (1957)



調整可能なパラメータ w

～ 各シナプスのつながりの強さ

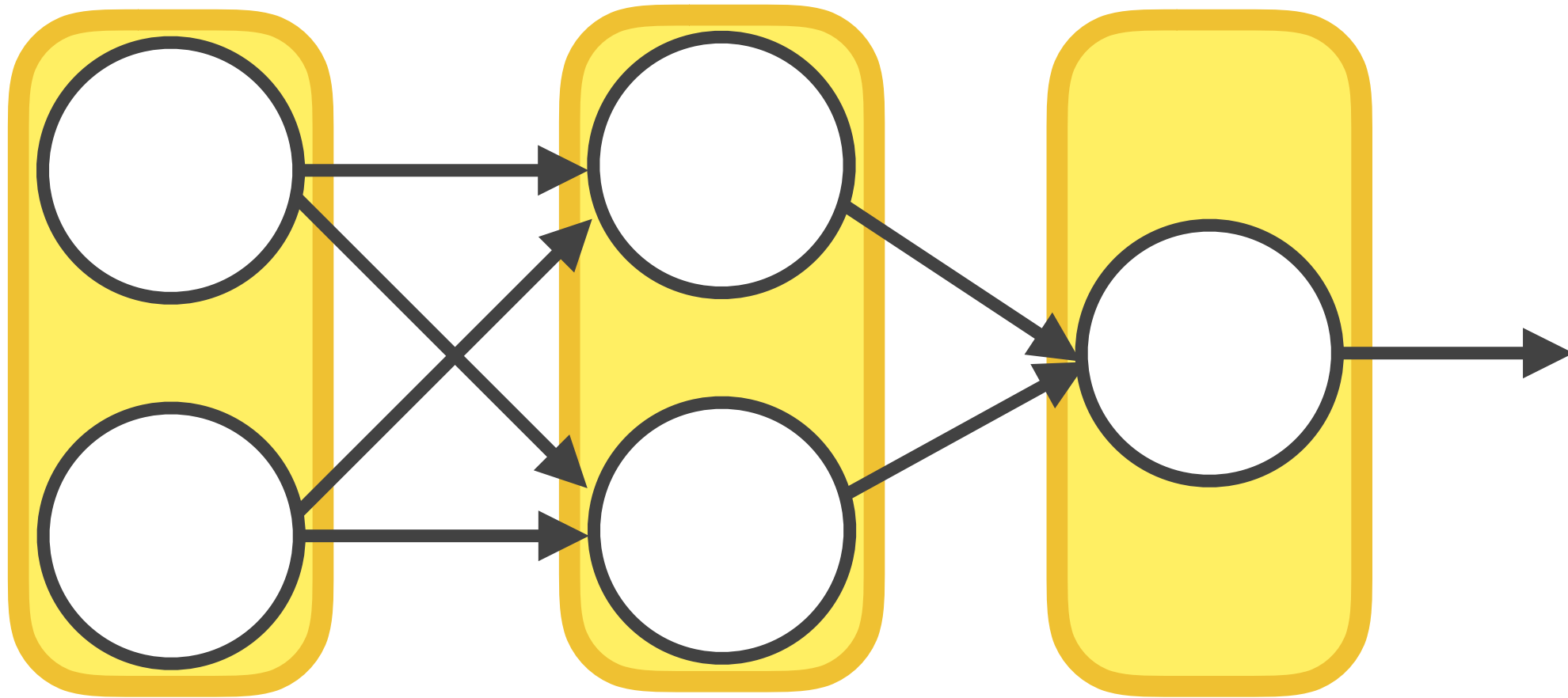
Rosenblattのパーセプトロン (1957)



(人工)ニューラルネット

- 大雑把にネットワークをデザインすれば良い
- 学習で性能が調整される

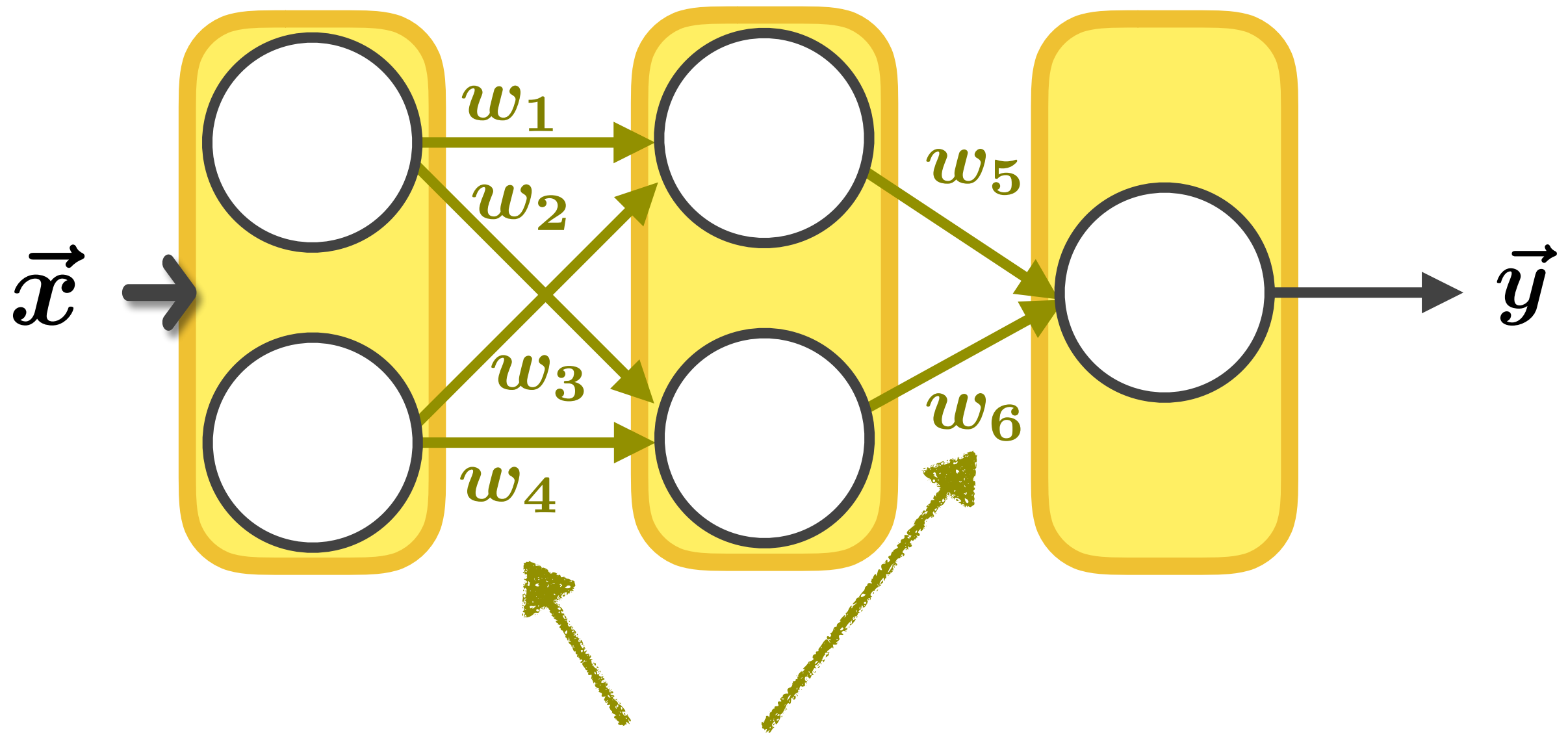
Rosenblattのパーセプトロン (1957)



多層ニューラルネット

- 層構造を持たせるのが実は重要
- 層数が多い = 深層

Rosenblattのパーセプトロン (1957)



これらパラメータを学習

板書でちょっと整理します

Rosenblattのパーセプトロン (1957)



<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/KI-und-Softcomputing/Neurales-Netz>

Rosenblattのパーセプトロン (1957)



<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/KI-und-Softcomputing/Neuronales-Netz>

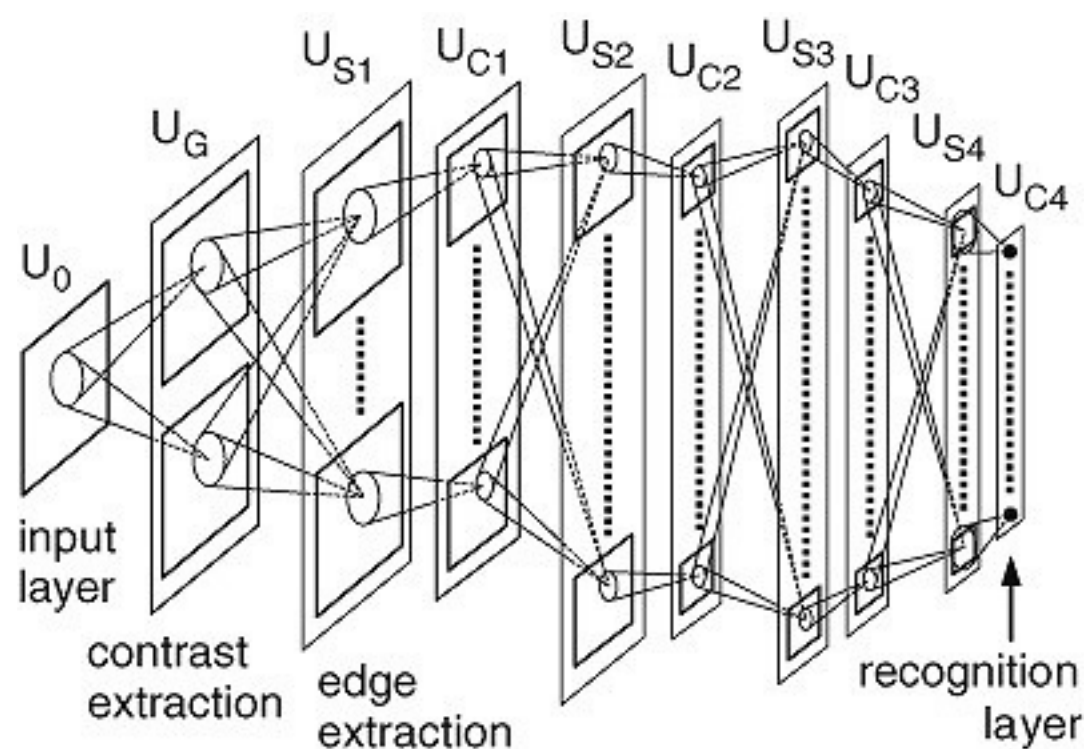
第一次AIブームを引き起こした
とも言われるが、若くしてロー
ゼンブラットが亡くなるのと同
時期にAI冬の時代へと突入

深層学習への道

ニューラルネットに視覚を持たせる (1979)

1979年 福島邦彦 (NHK放送科学基礎研究所)

ネオコグニトロンを提唱



ニューラルネットを学ばせる (1965-)

1965年 甘利俊一（九州大学）

ニューラルネットの学習法を初めて提唱

<http://www.riken.jp/pr/topics/2012/20121031/>



1986年 ラメルハート（カルフォルニア大学）ら

バックプロパゲーションとして再発見

**Learning representations
by back-propagating errors**

**David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams***

* Institute for Cognitive Science, C-015, University of California,
San Diego, La Jolla, California 92093, USA

† Department of Computer Science, Carnegie-Mellon University,
Pittsburgh, Philadelphia 15213, USA



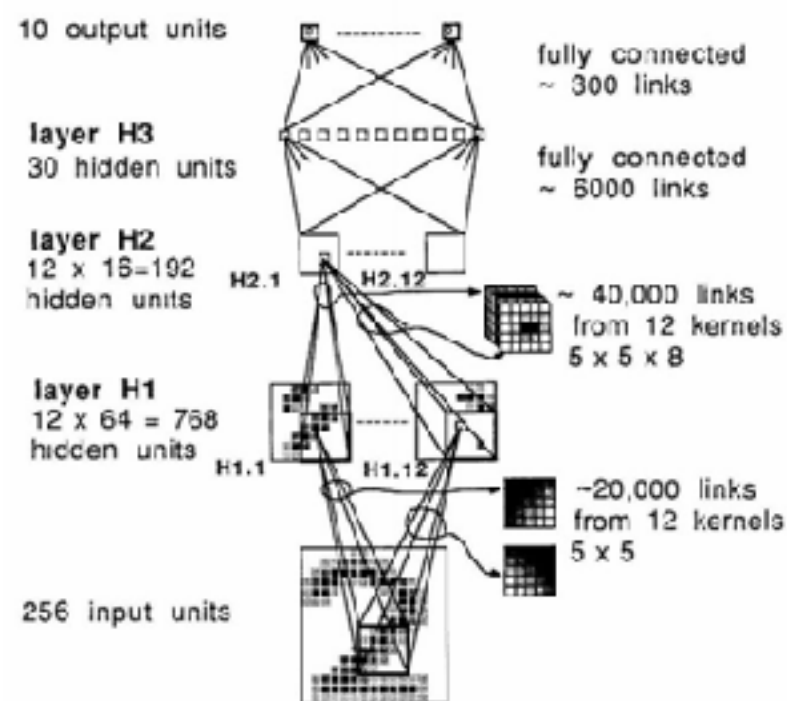
<https://news.stanford.edu/news/2011/march/david-rumelhart-obit-031711.html>

複雑な視覚ネットワークを学ばせる (1989)

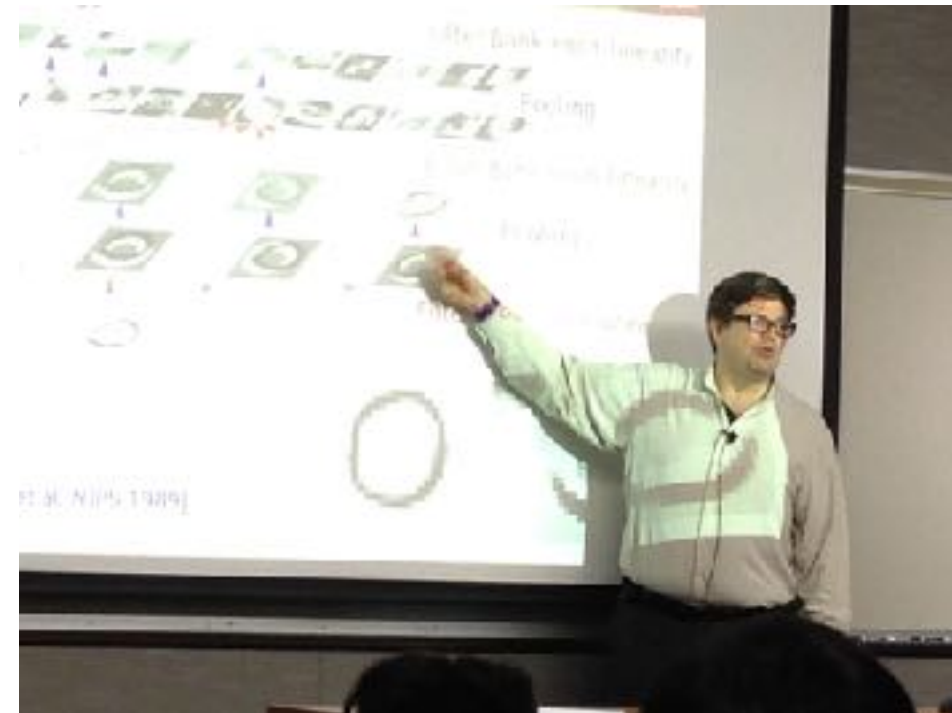
1989年 LeCun (AT&T Bell研究所)

LeNetを提唱、バックプロパゲーションを適用

多層化に成功し、手書き数字を読ませることに成功



Backpropagation applied to handwritten zip code recognition,
Lecun et al., 1989



https://en.wikipedia.org/wiki/Yann_LeCun#/media/File:Yann_LeCun_at_the_University_of_Minnesota.jpg

現在の深層学習はLeNetの直系の子孫 (eg. GoogLeNet)

90年代から2000年初頭



ニューラルネット氷河期…



ヒントンの先生をかけた研究の開花



http://www.macleans.ca/wp-content/uploads/2016/03/MAR19_HINTON_POST01.jpg

Geoffrey Everest Hinton

ニューラルネットのブームを引き起こす研究を初期からしてきたの
みならず、冬の時代（90年代）もずっと地道に研究を続けた

AlexNetとして結実 (2012)

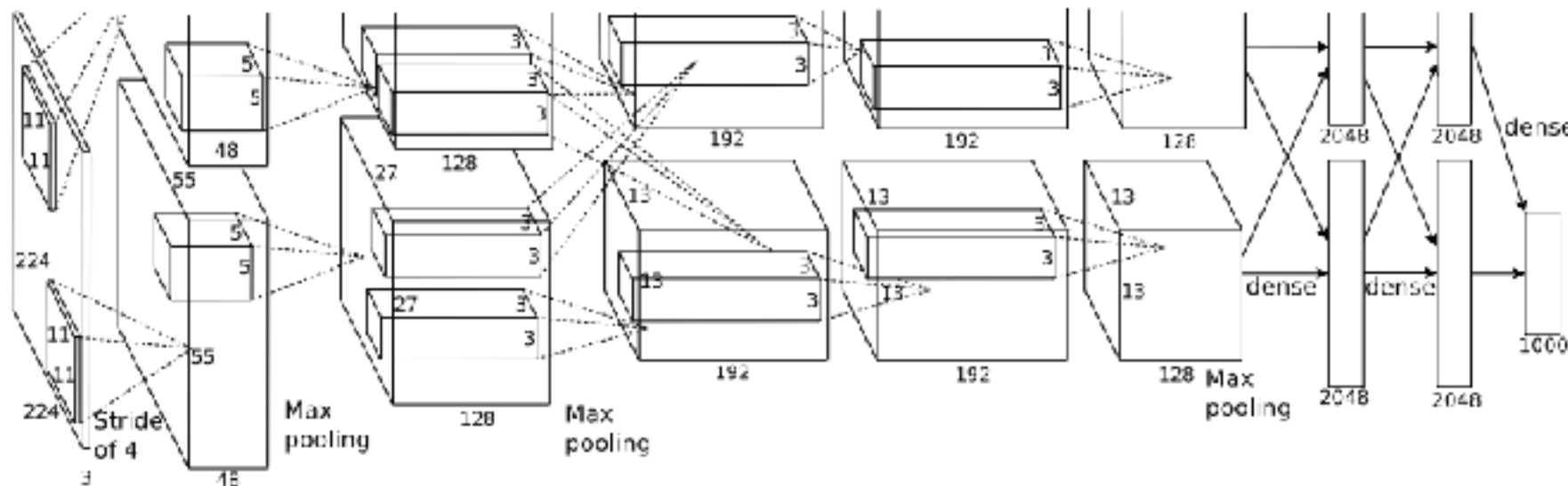
2012年 Krizhevsky, Sutskever, Hinton (トロント大学)

深層学習技術の確立

驚異的な画像認識性能



https://www.wired.com/2013/03/google_hinton/



- 画像認識
- 画像生成
- 自動翻訳
- 音声認識
- 音声合成
- 自動運転

...

あなたの携帯電話にも。

2012年から深層学習としてニューラルネットが大発展。

科学技術や産業を変えつつある

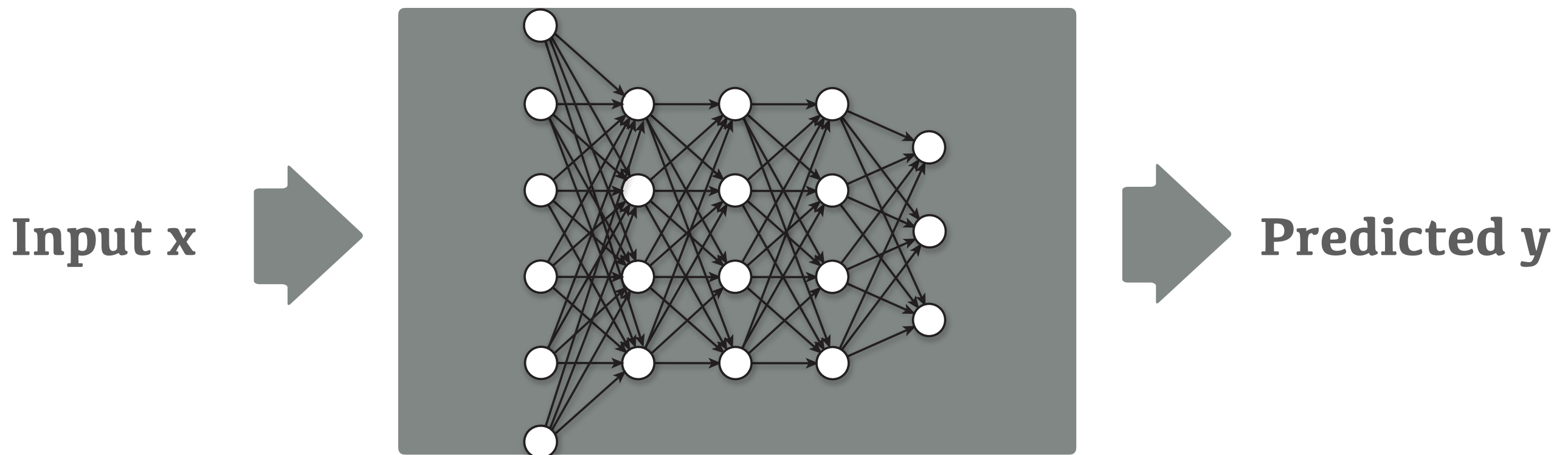


この章のまとめ

深層学習(Deep Learning)

深層学習は、マシン=モデルを**グラフ**を使って直感的に設計できる便利な手法

この**グラフ**は、我々の脳(神経系)を模倣したもの

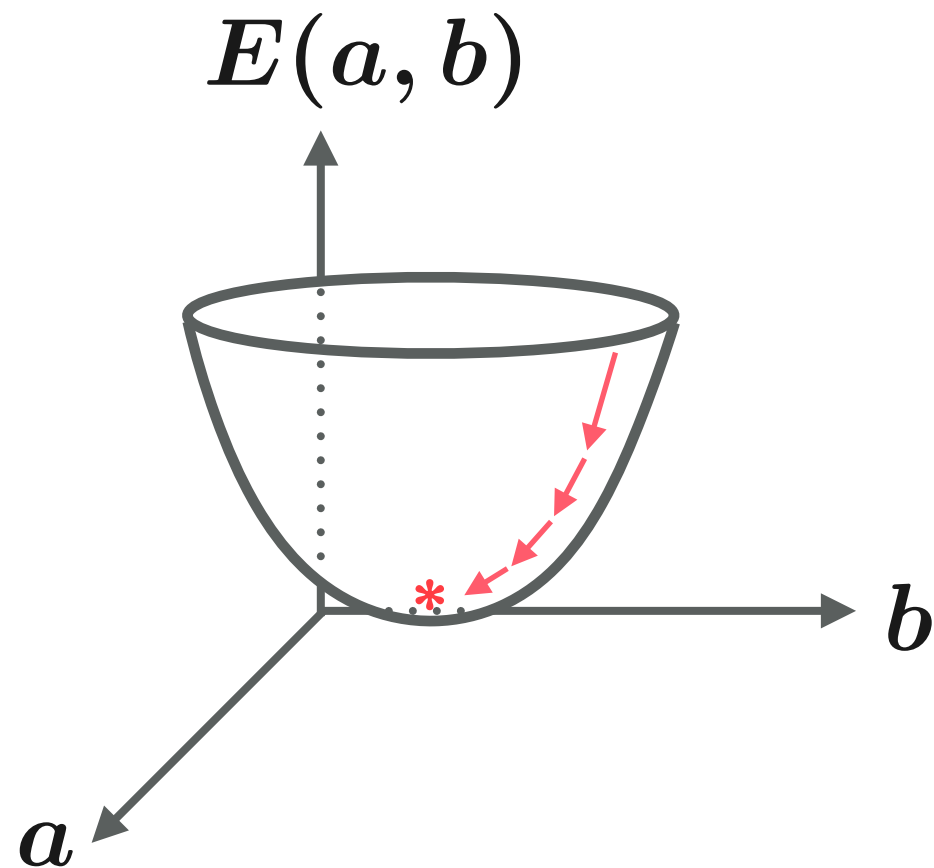


2. 学習の実際

勾配降下法による最適化

勾配降下法

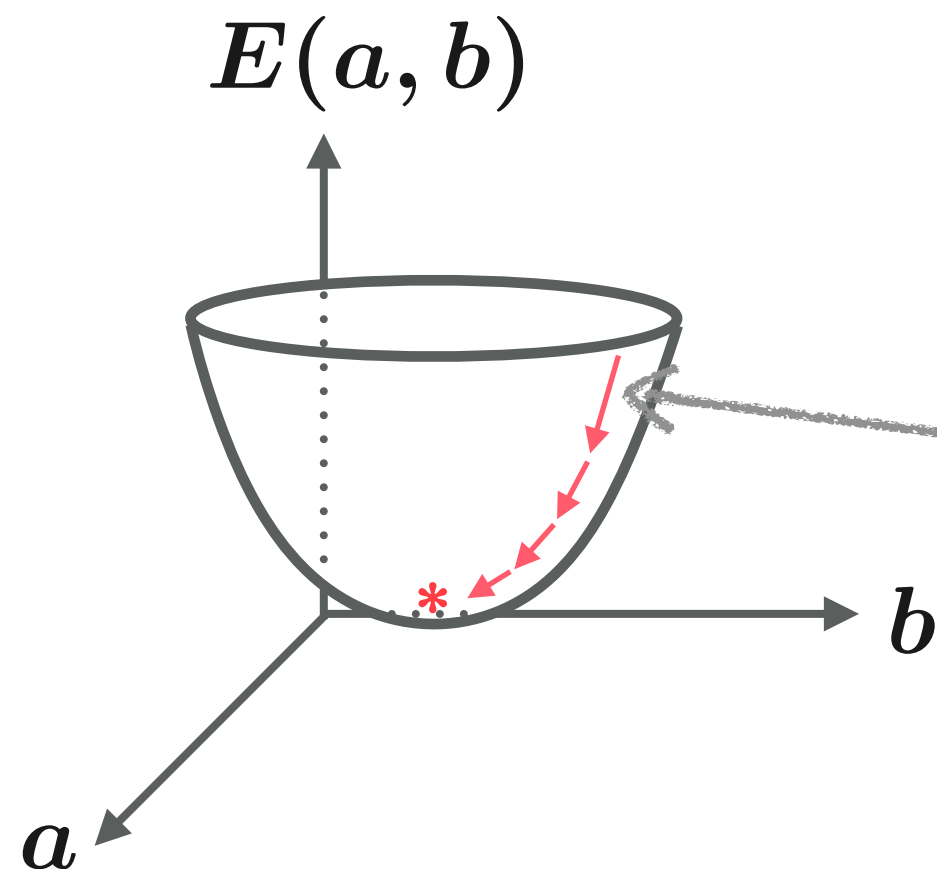
最適化(最小化)問題を数値的に解く



坂道を転がすように動かしていけば、やがて底で止まる

勾配降下法

最適化(最小化)問題を数値的に解く



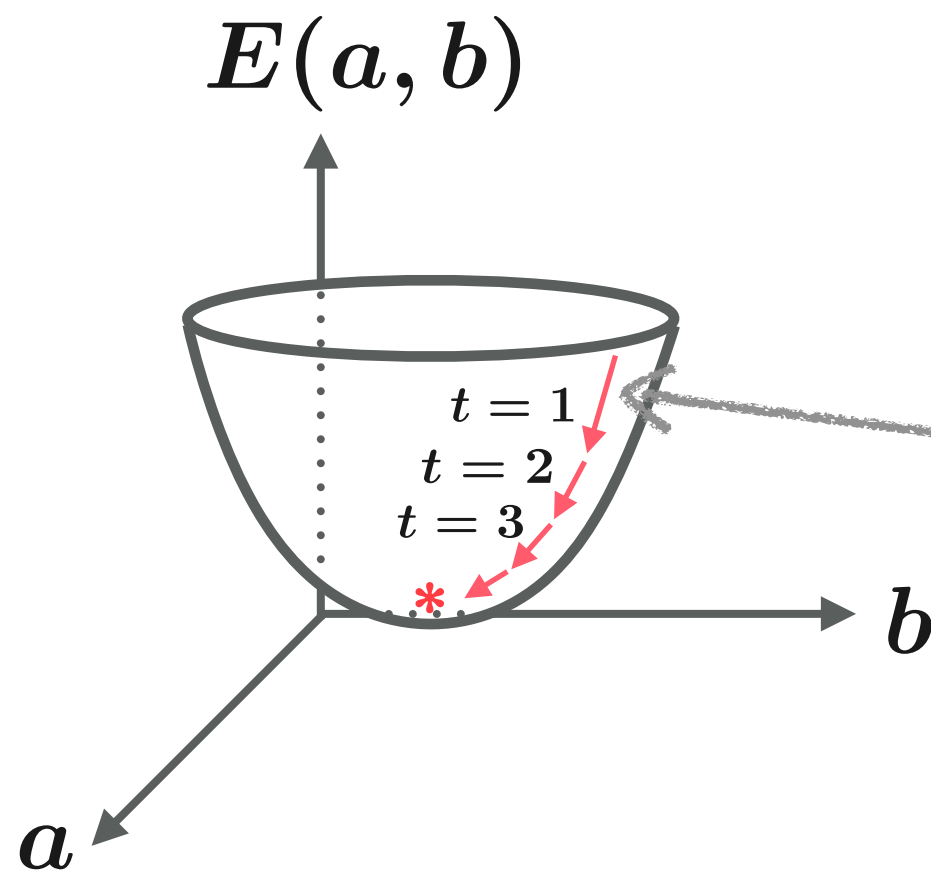
坂道を転がすように動かしていけば、やがて底で止まる

$$-\begin{pmatrix} \frac{\partial E}{\partial a} \\ \frac{\partial E}{\partial b} \end{pmatrix}$$

勾配 gradient

勾配降下法

最適化(最小化)問題を数値的に解く



坂道を転がすように動かしていけば、やがて底で止まる

$$-\begin{pmatrix} \frac{\partial E}{\partial a} \\ \frac{\partial E}{\partial b} \end{pmatrix}$$

勾配降下

gradient descent

$$a^{t+1} = a^t - 0.01 \frac{\partial E(a^t, b^t)}{\partial a}$$

$$b^{t+1} = b^t - 0.01 \frac{\partial E(a^t, b^t)}{\partial b}$$

ニューラルネットワーク のオンライン学習

ミニバッチ学習

単回帰の例

$$E(a, b) = \frac{1}{N} \sum_{n=1}^N \left(\hat{y}(x_n : a, b) - y_n \right)^2$$

ミニバッチ学習

単回帰の例

$$E(a, b) = \frac{1}{N} \sum_{n=1}^N \left(\hat{y}(x_n : a, b) - y_n \right)^2$$

勾配法を使うときは、各時刻 t で毎回わざわざ全部のデータに関して平均を取らなくても良い。ランダムに一部だけデータを取ってきて(部分集合=ミニバッチ)、それに関して誤差関数を計算

ミニバッチ学習

単回帰の例

$$E(a, b) = \frac{1}{N} \sum_{n=1}^N \left(\hat{y}(x_n : a, b) - y_n \right)^2$$



$$E^t(a, b) = \frac{1}{M} \sum_{m=1}^M \left(\hat{y}(x_m : a, b) - y_m \right)^2$$

毎時刻、ランダムに部分集合を選び直す

$$\{(x_m, y_m)\}_{m=1}^M \subset \{(x_n, y_n)\}_{n=1}^N$$

オンライン学習

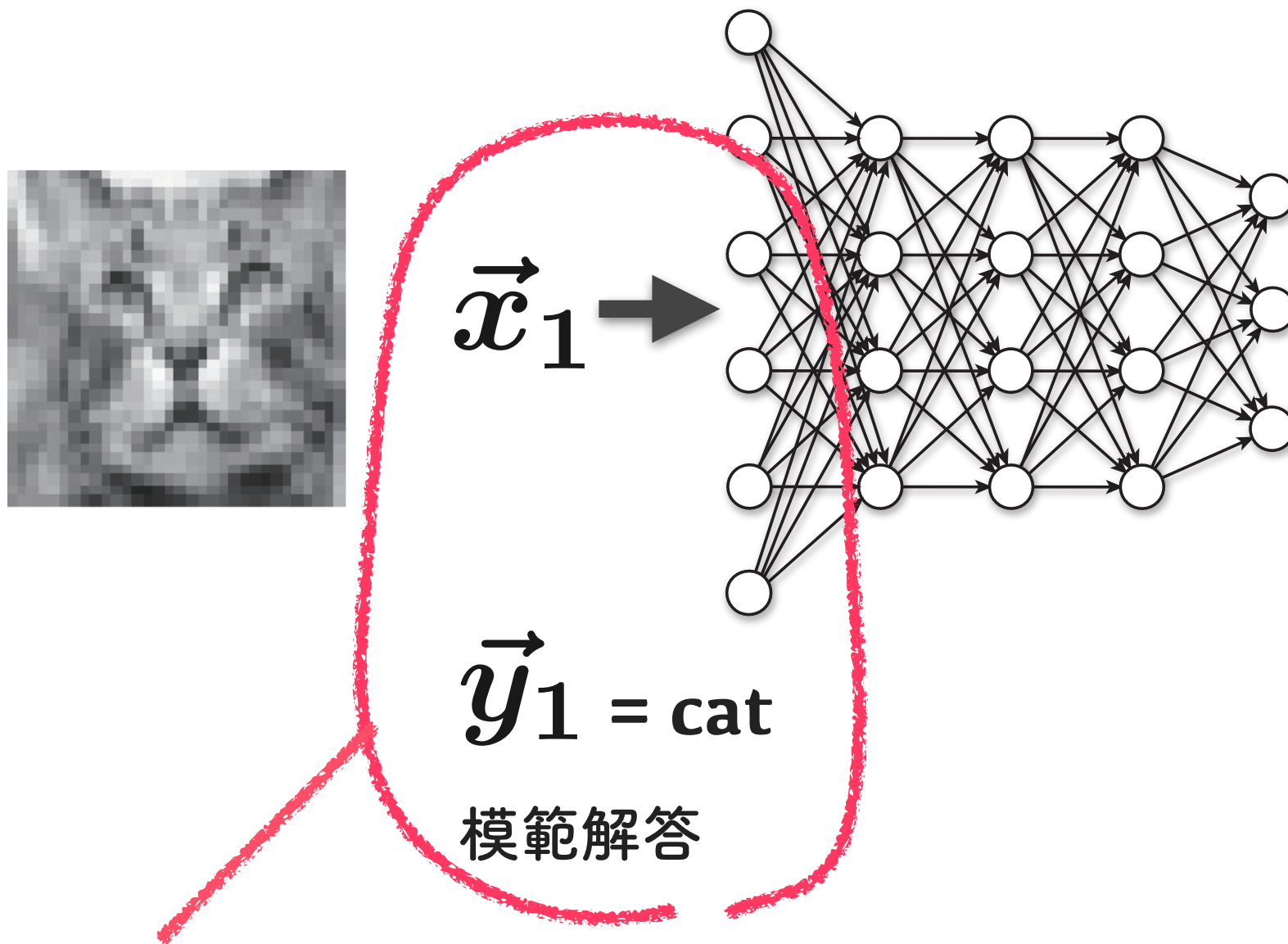
極端な場合は、ミニバッチがデータ点一個だけ
($M=1$)

$$E^t(a, b) = \frac{1}{M} \sum_{m=1}^M \left(\hat{y}(x_m : a, b) - y_m \right)^2$$

毎時刻、ランダムに部分集合を選び直す

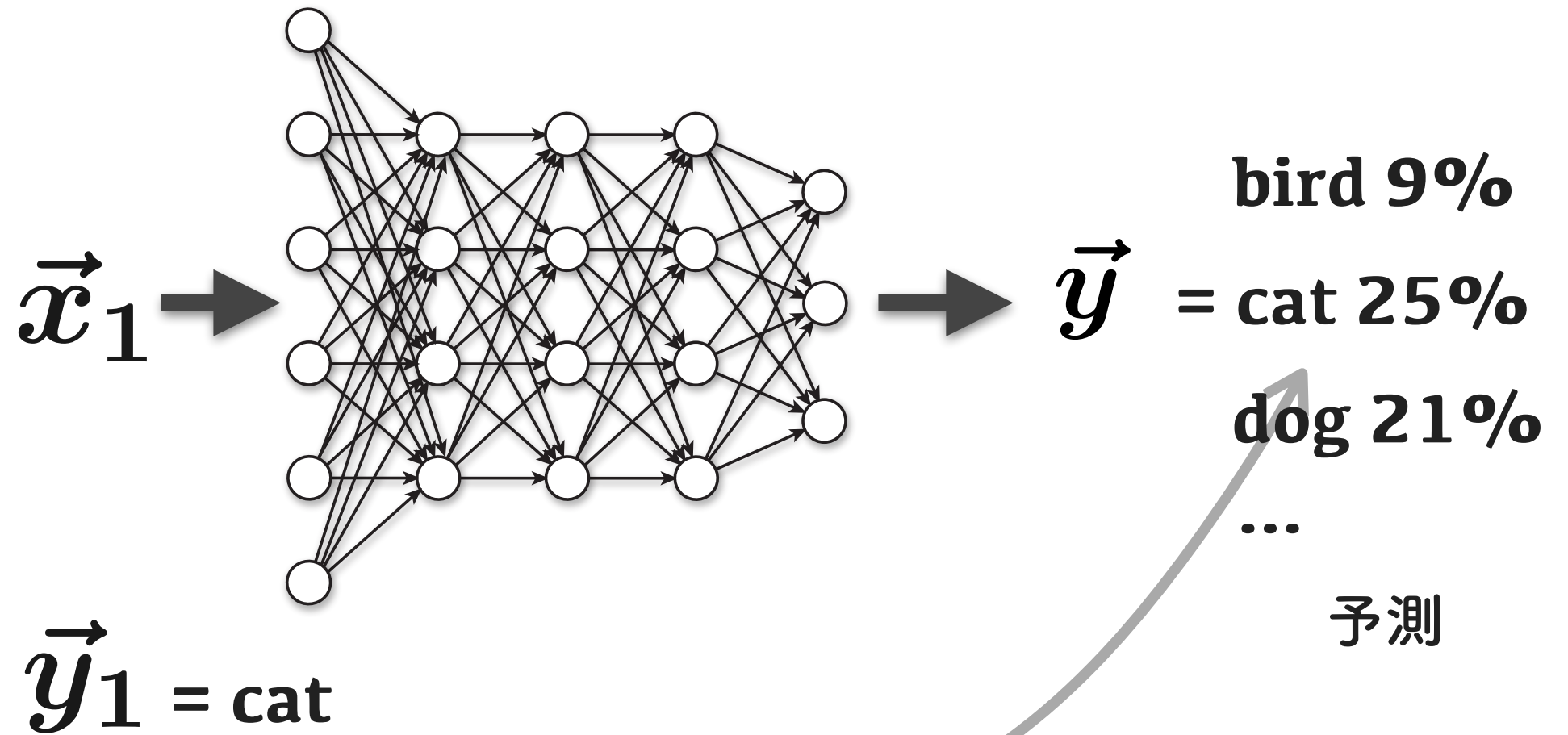
$$\{(x_m, y_m)\}_{m=1}^M \subset \{(x_n, y_n)\}_{n=1}^N$$

オンライン学習



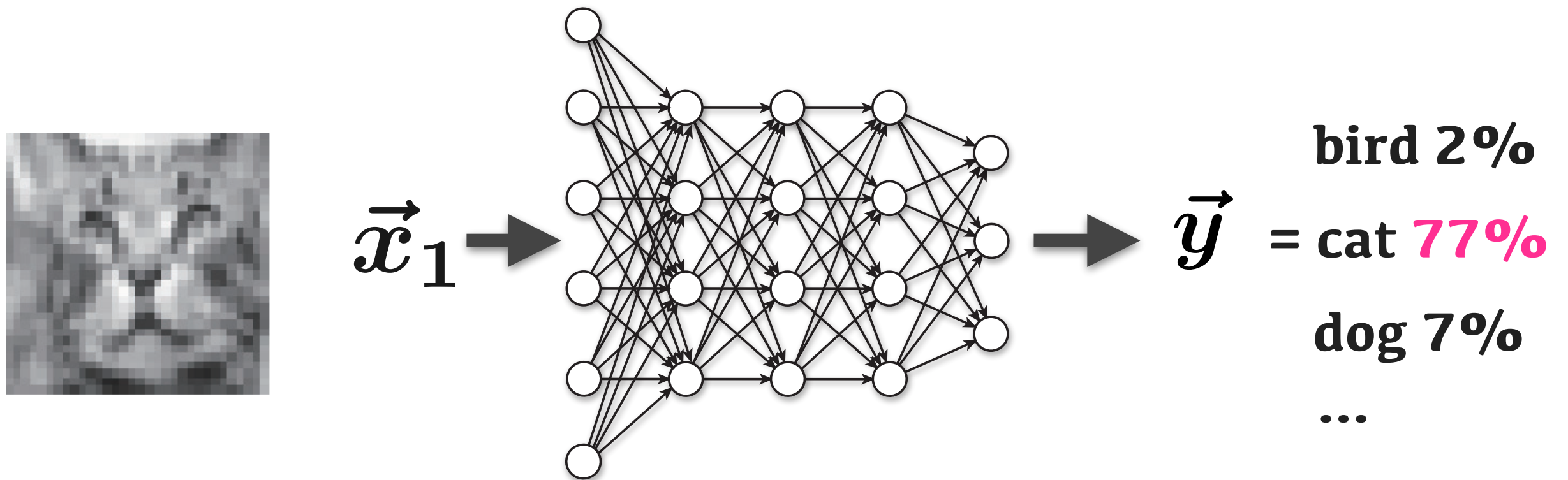
訓練データの中の一つの事例をランダムに取ってくる

オンライン学習



はじめは当然、予測性能は全く良くない

オンライン学習

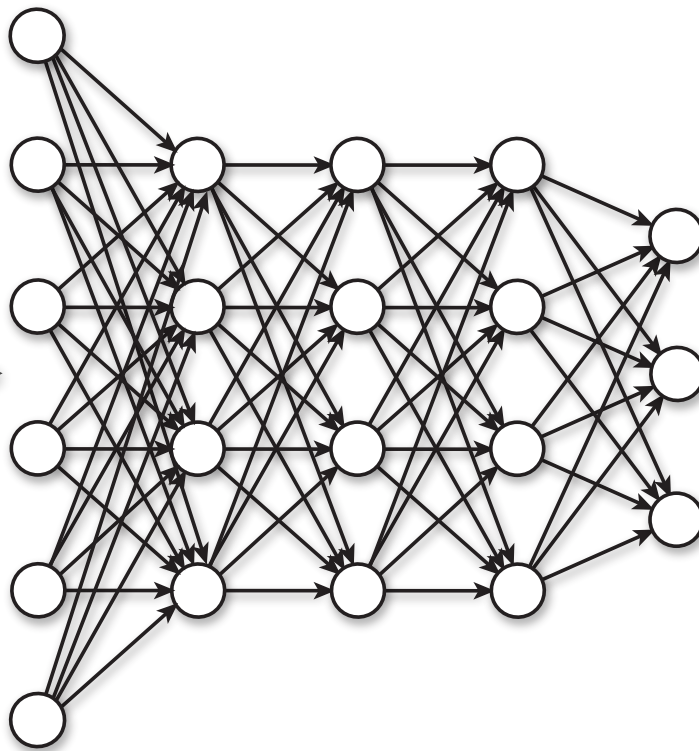


「予測の精度が上がるように」 = 「誤差関数が小さくなるように」 パラメータを一回だけ勾配降下

オンライン学習



$\vec{x}_2$

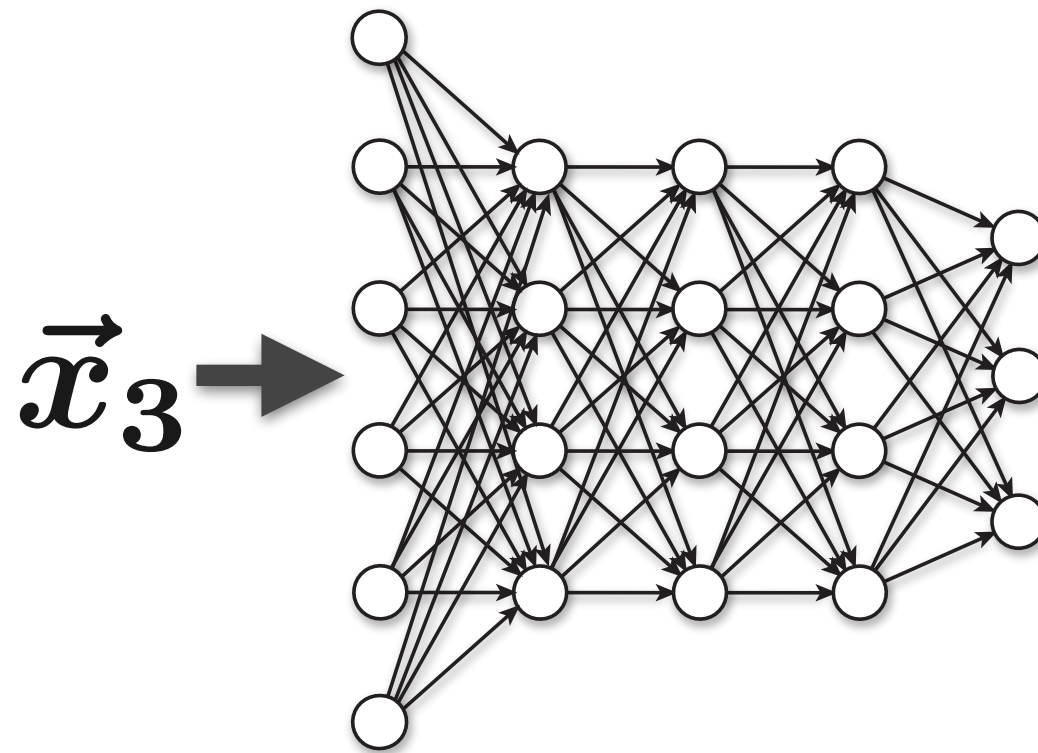


$\vec{y}_2 = \text{bird}$

$\vec{y}$ = bird 12%
= cat 13%
dog 19%
...

予測

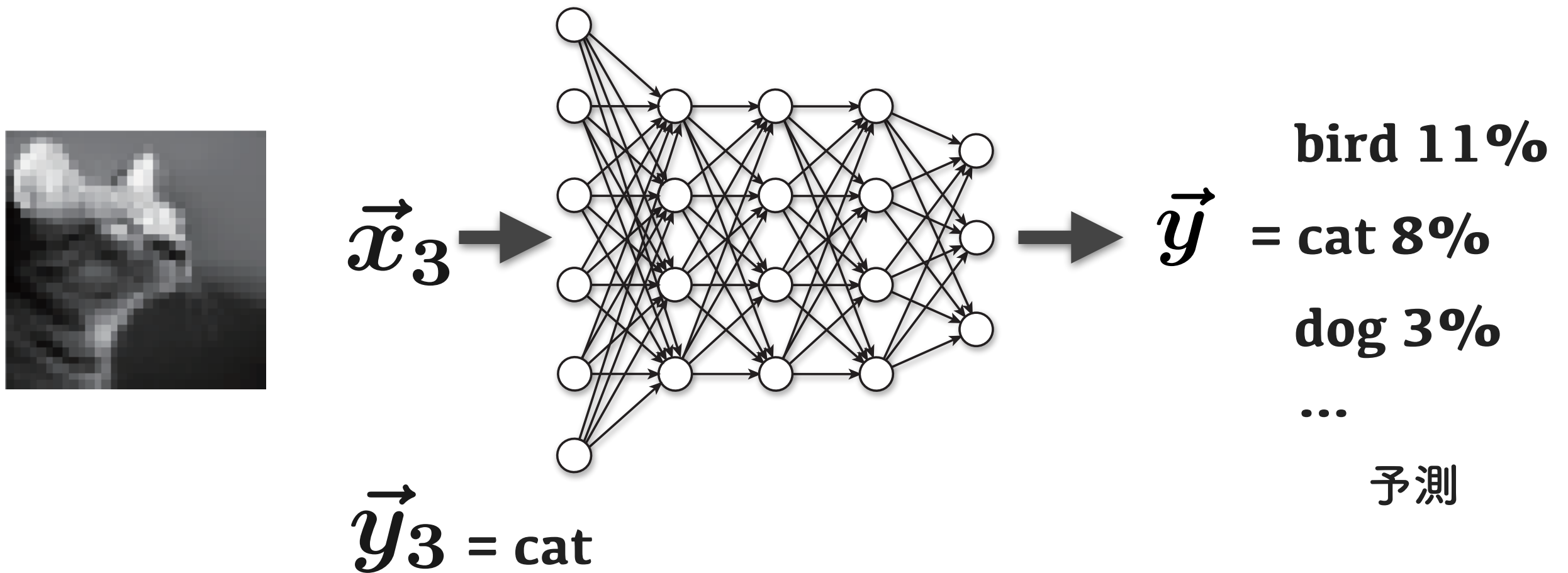
オンライン学習



$\vec{y}_3 = \text{cat}$

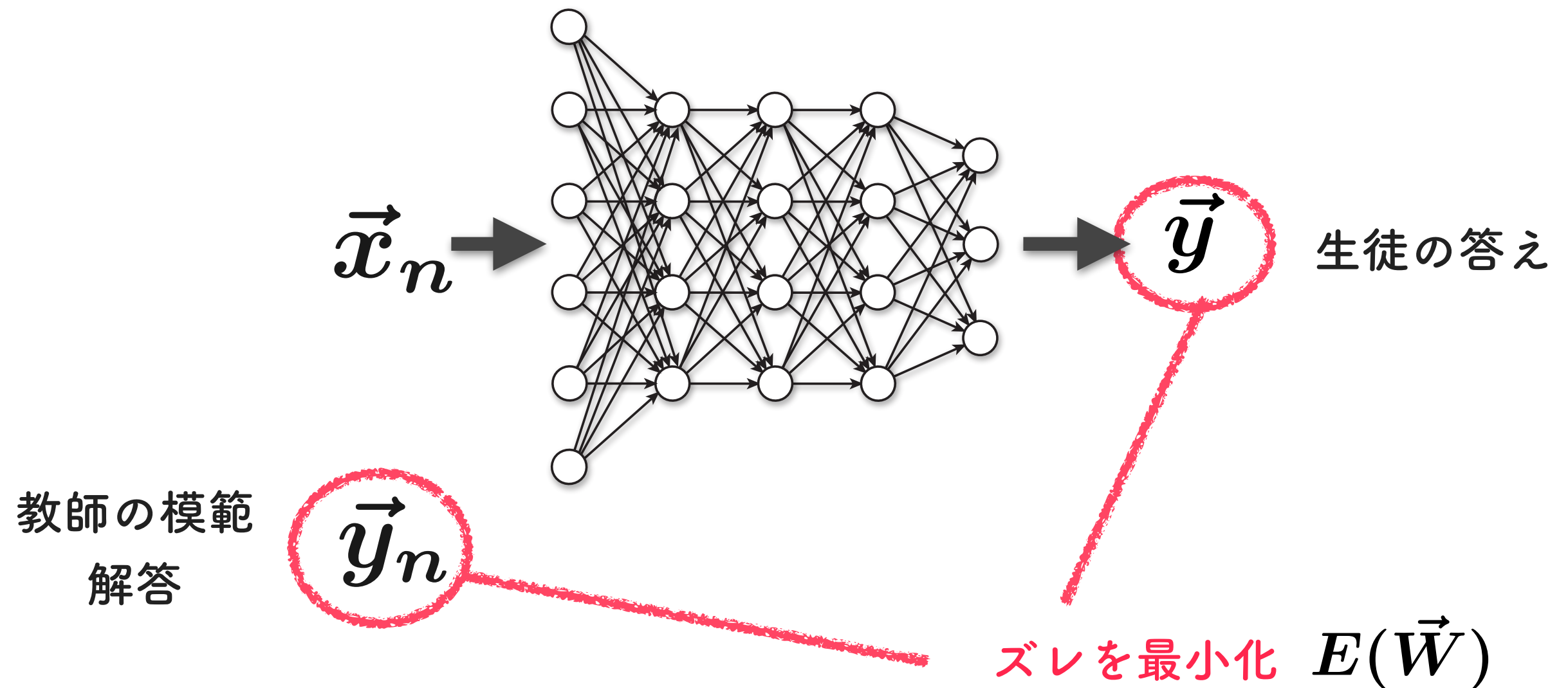
$\vec{y}$ = bird 11%
= cat 8%
dog 3%
...
予測

オンライン学習



これをかなりの回数繰り返す（例えば10万回）

教師あり学習 Supervised Learning



xに対してyが用意されている場合は、このような
教師あり学習ができる

続きは次回