


**第1回**  
 **$y=ax+b$ から始める**  
**初心者向けML講座**



# 第1回 $y=ax+b$ から始める 初心者向けML講座

なお、 $y=ax+b$ が出てくるのは後半です

# 自己紹介

名前：Earl Klutz（読み方：アール・クルツ）

- 元はPC・サーバ管理運用、ネットワークインフラ方面生まれ
  - 十数年ほど大手企業3社+a、放浪しつつITと名の付くもの一通りやる。
  - PC5000台管理、サーバ100台管理、ビル1棟のネットワーク配線、プログラミング開発(バック～フロントエンド全部)など色々やる。
  - 何故かAndroidアプリも開発する羽目になった事も…
  - F社方面で仮想化技術（今で言うクラウド）の研究などもしていました。
- 一時期は会社を立てたりもしましたが、現在はフリーで運用・設計・開発など何でもやる便利屋さんです。
- 週5で働くのは微妙 & 請負開発は忙しい時と暇な時の差が激しいので企業向けのPG講師も始める。
  - 気が付いたらJava、PHP、Python、ABAP、SQL、Linux、AWS、Ciscoルータなど色々教える羽目に。
- AIも教えてくれと言われJDLA（日本ディープラーニング協会）のE資格を取得。
- 2020年ごろからとある大学教授に作ってもらった資料を基にE資格講座を全部作って教えることに。

教えてくれ。俺たちははあと何個のプログラム言語を教えればいい？

AIとは？

# AIとは？

- あたかも人間のようにふるまうコンピューターの事。
- 未知を予測するコンピュータ。
- 人工的に作ら有れた知能や心を持つメカ。

→など、色々と言われており、あまりはっきりと定義はされいません。

## 何故か？

- 時代と共にAIと呼ばれるものの性質が変わっている為。  
→古くはインベーダーゲームの敵もAIと呼んでいたこともある。

• AIの進化が早すぎて定義そのものが変わっている！！

つまり？

- AIの成長速度が



でもどうしても定義しておくとする？

「人工的につくられた人間のような知能、ないしはそれをつくる技術」

By 東京大学 松尾 豊教授

分かりやすく（乱暴に）言えば

人間っぽい動きをする人工物は何でもAI

と言えなくもない

# ではどういう技術があるのか？

- レベル1

シンプルな制御プログラム（センサーなどを用いた単純な動きをするもの）

例：エアコン、洗濯機など

- レベル2

古典的な人工知能（学習はしないがある程度複雑な動きをするもの）

例：ゲームのAI、お掃除ロボットなど

- レベル3

機械学習を用いた人工知能（処理の中で学習機能を持つもの？）

例：通販サイトのおすすめ商品、迷惑メール判定、若干古典的だがドラクエ4(1990年発売)の仲間のAIなど

- レベル4

ディープラーニングを用いた人工知能

例：Siri、ChatGTP、stable diffusionなど

最初はザラキしかしないクリフトも学習してちゃんと戦うようになるのでこの分類か？  
(ただし敵の種類ごとに学習するのでほぼ無意味ですが)



# ではどういう技術があるのか？

- レベル1

シンプルな制御プログラム（センサーなどを用いた単純な動きをするもの）

例：エアコン、洗濯機など

- レベル2

古典的な人工知能（学習はしないがある程度複雑な動きをするもの）

例：ゲームのAI、お掃除ロボットなど

- レベル3

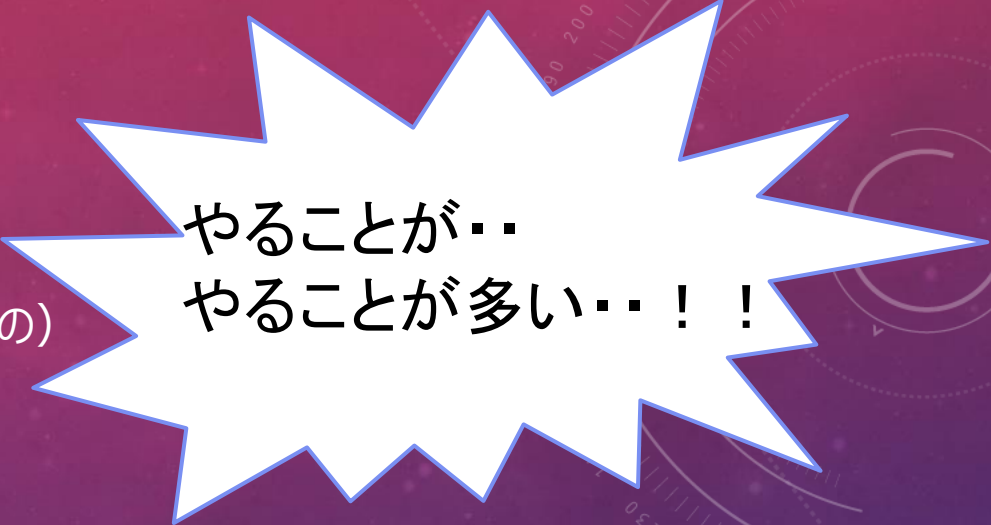
機械学習を用いた人工知能（処理の中で学習機能を持つもの？）

例：通販サイトのおすすめ商品、迷惑メール判定、若干古典的だがドラクエ4(1990年発売)の仲間のAIなど

- レベル4

ディープラーニングを用いた人工知能

例：Siri、ChatGTP、stable diffusionなど



やることが..  
やることが多い...!!

# ではどういう技術があるのか？

- レベル1

シンプルな制御プログラム（センサーなどを用いた単純な動きを

例：エアコン、洗濯機など

- レベル2

古典的な人工知能（学習はしないがある程度複雑な動きをするか

例：ゲームのAI、お掃除ロボットなど

- レベル3

機械学習を用いた人工知能（処理の中で学習機能を持つもの？）

例：通販サイトのおすすめ商品、迷惑メール判定、若干古典的だがドラクエ4(1990年発売)の仲間のAIなど

- レベル4

ディープラーニングを用いた人工知能

例：Siri、ChatGTP、stable diffusionなど

講座としての本題はレベル4  
ただし、ディープラーニングは機械学習  
の1分野なのでレベル3も含まれます。

# 最も基本的なディープラーニング

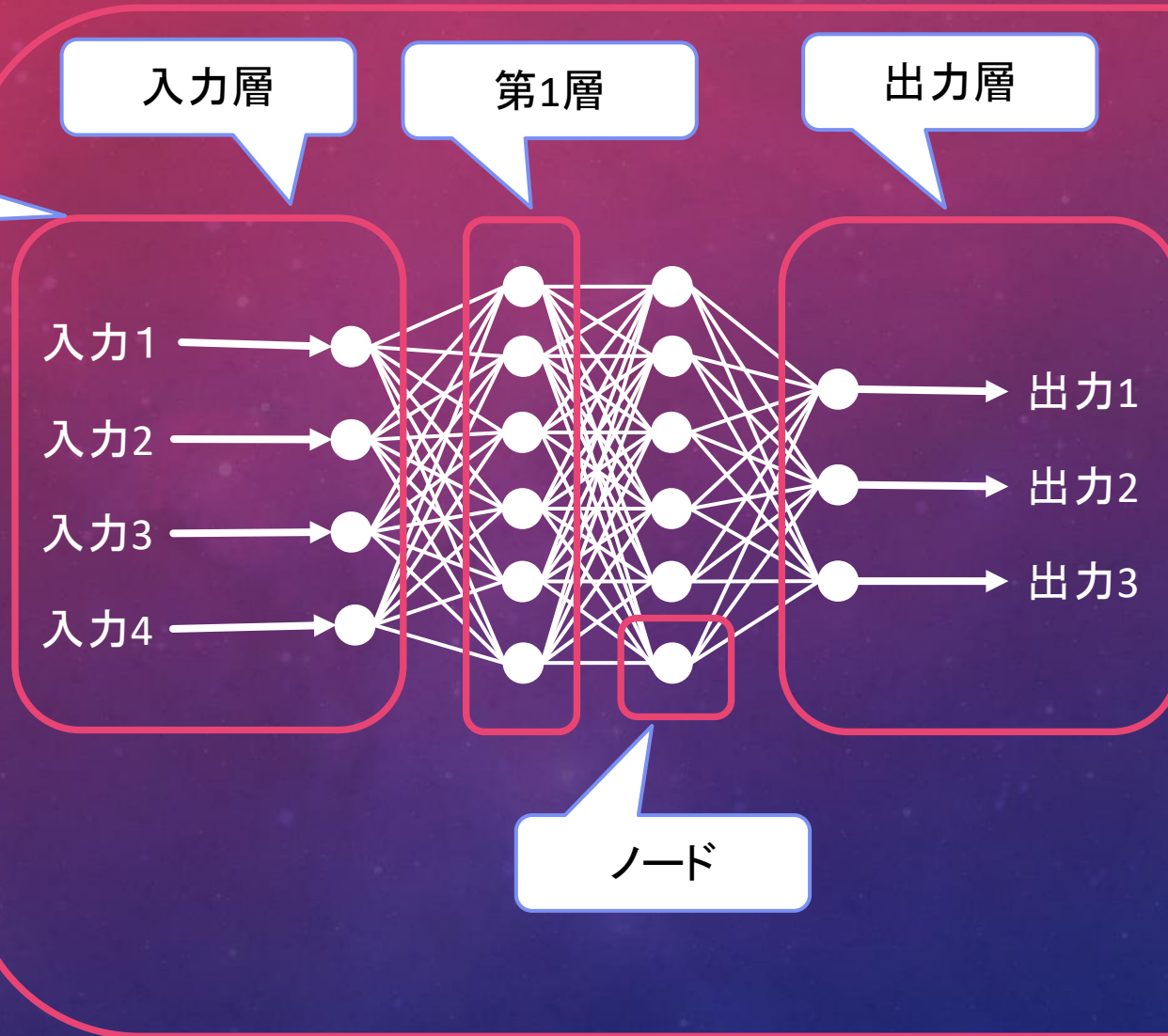
# ディープラーニングの基本はニューラルネットワーク

ML集会のアイコンにも書かれている



# ディープラーニングの基本はニューラルネットワーク (NN)

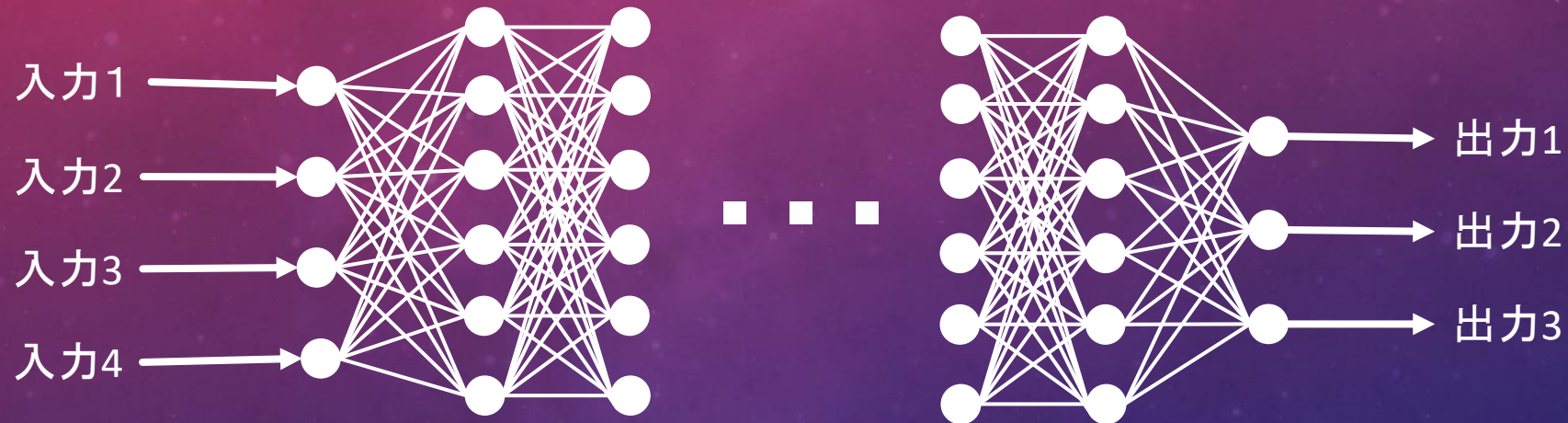
1個の●をノード  
ノード1列分で層  
と呼びます



入力から出力までの1つの  
ニューラルネットワークを  
モデルと呼びます

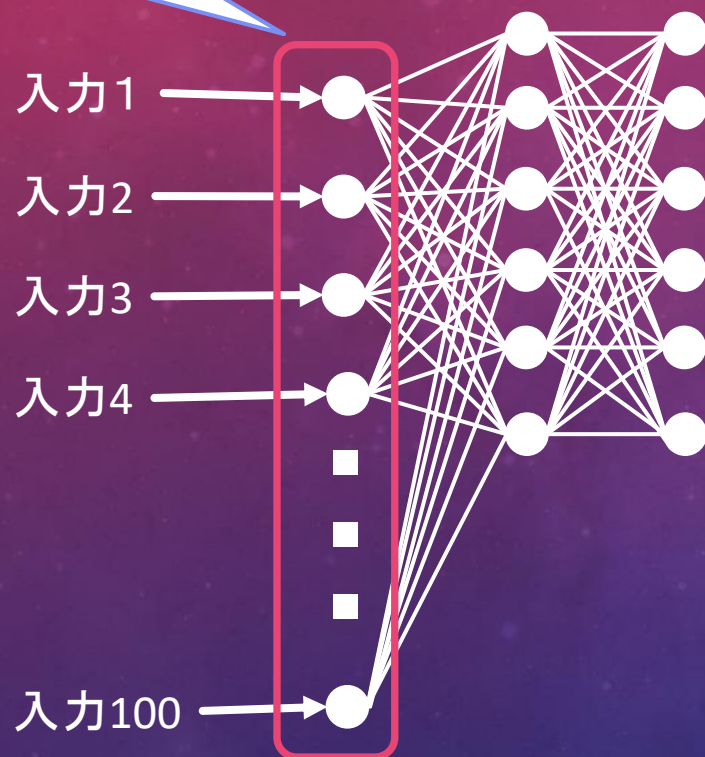
# 層を深くするとディープなニューラルネットワーク

- ニューラルネットワーク **≡** ディープラーニング

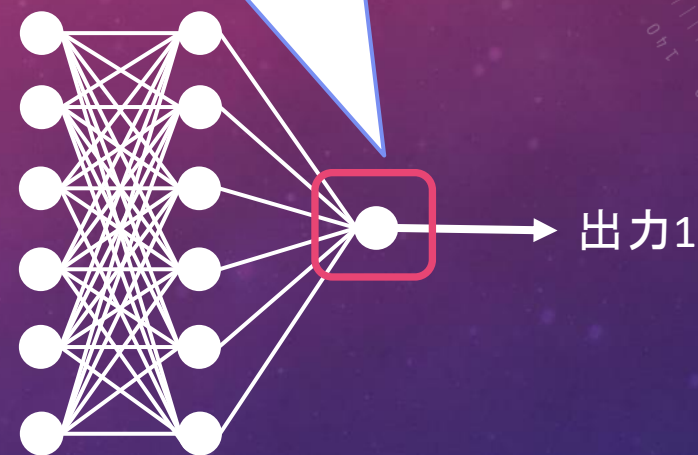


# 入力や出力のノードを増減させれば様々なものに対応

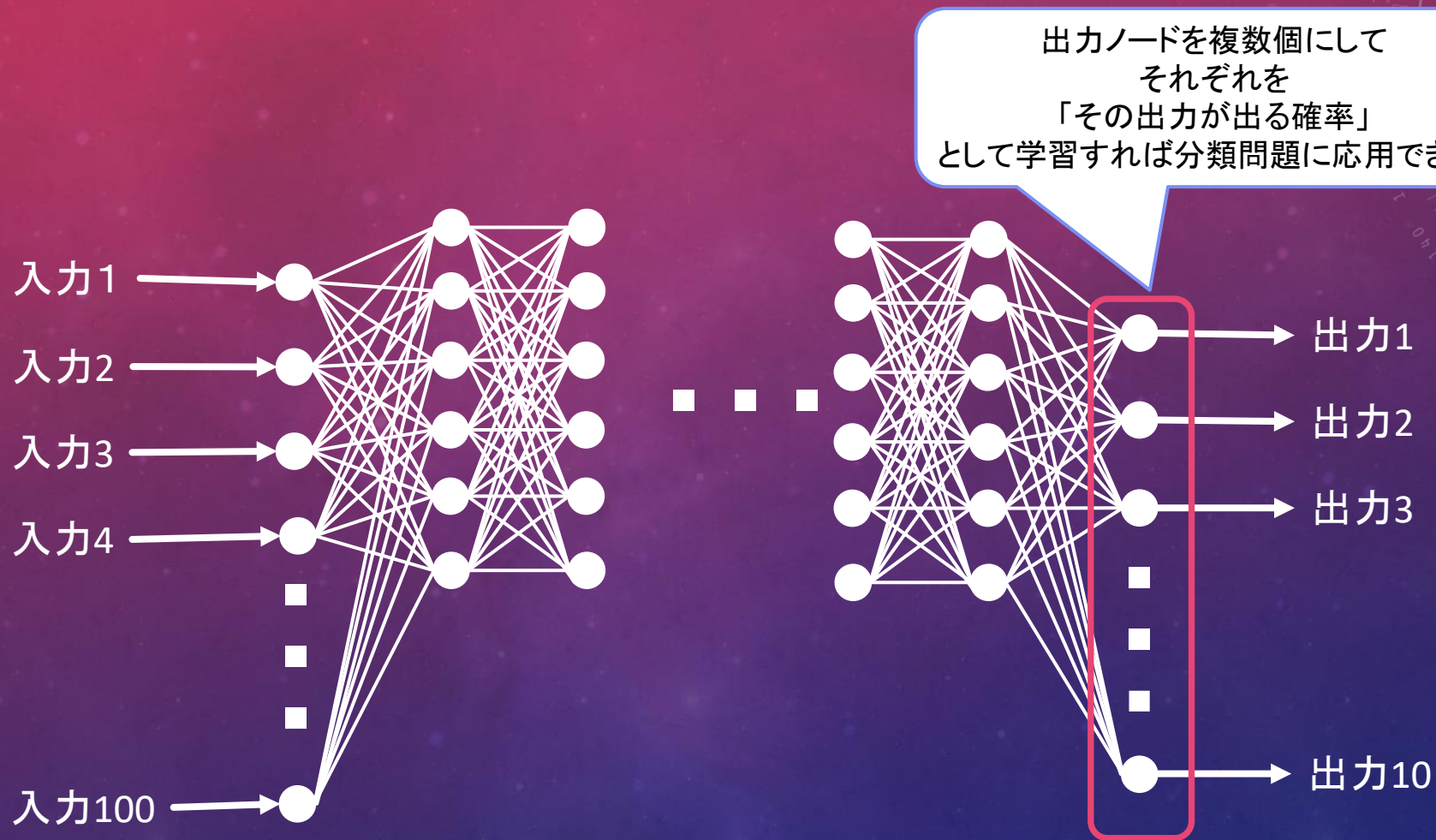
入力ノードを100個にすれば  
10×10pxのモノクロ画像を  
1ドットずつ入力出来る。



出力ノードを1つにすれば  
出力結果を何らかの  
数値予測データとして  
学習することができる



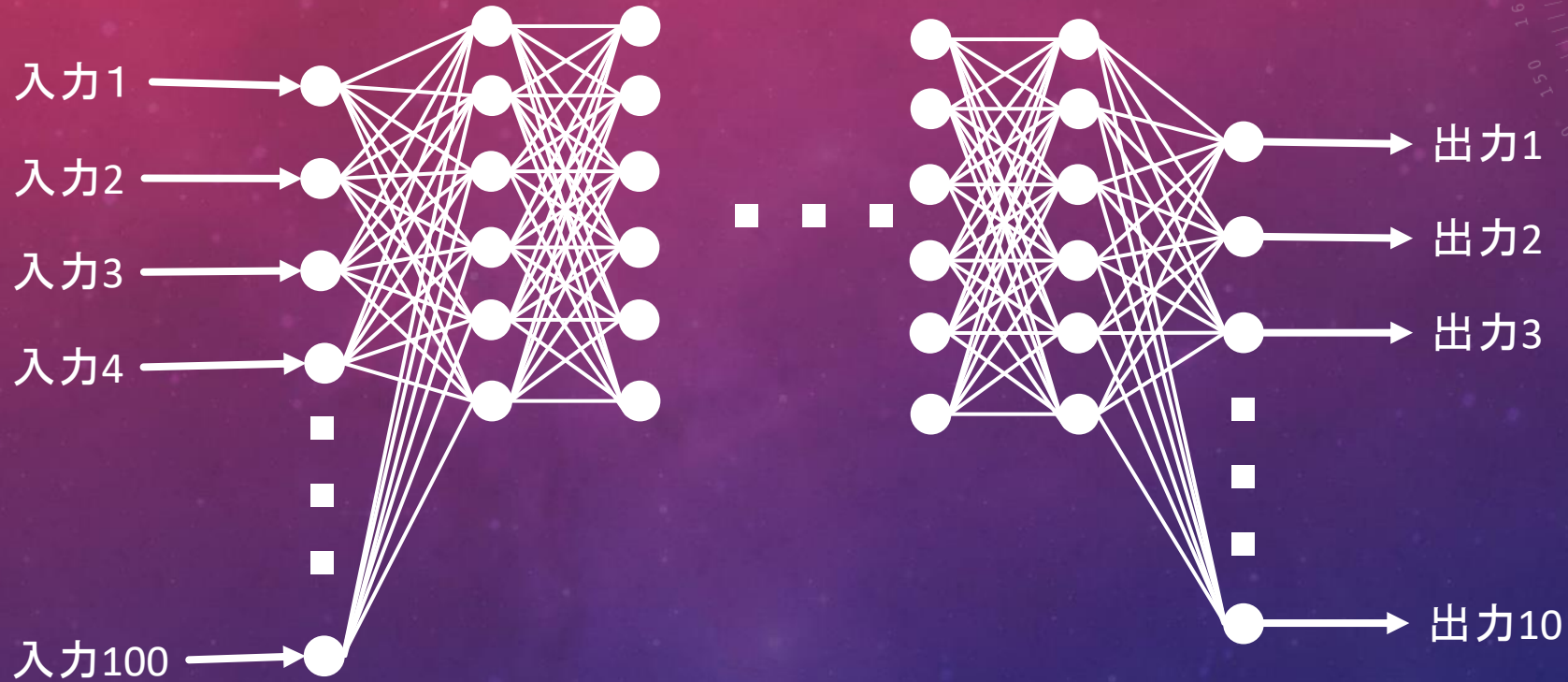
# 入力や出力のノードを増減させれば様々なものに対応





# つまり？

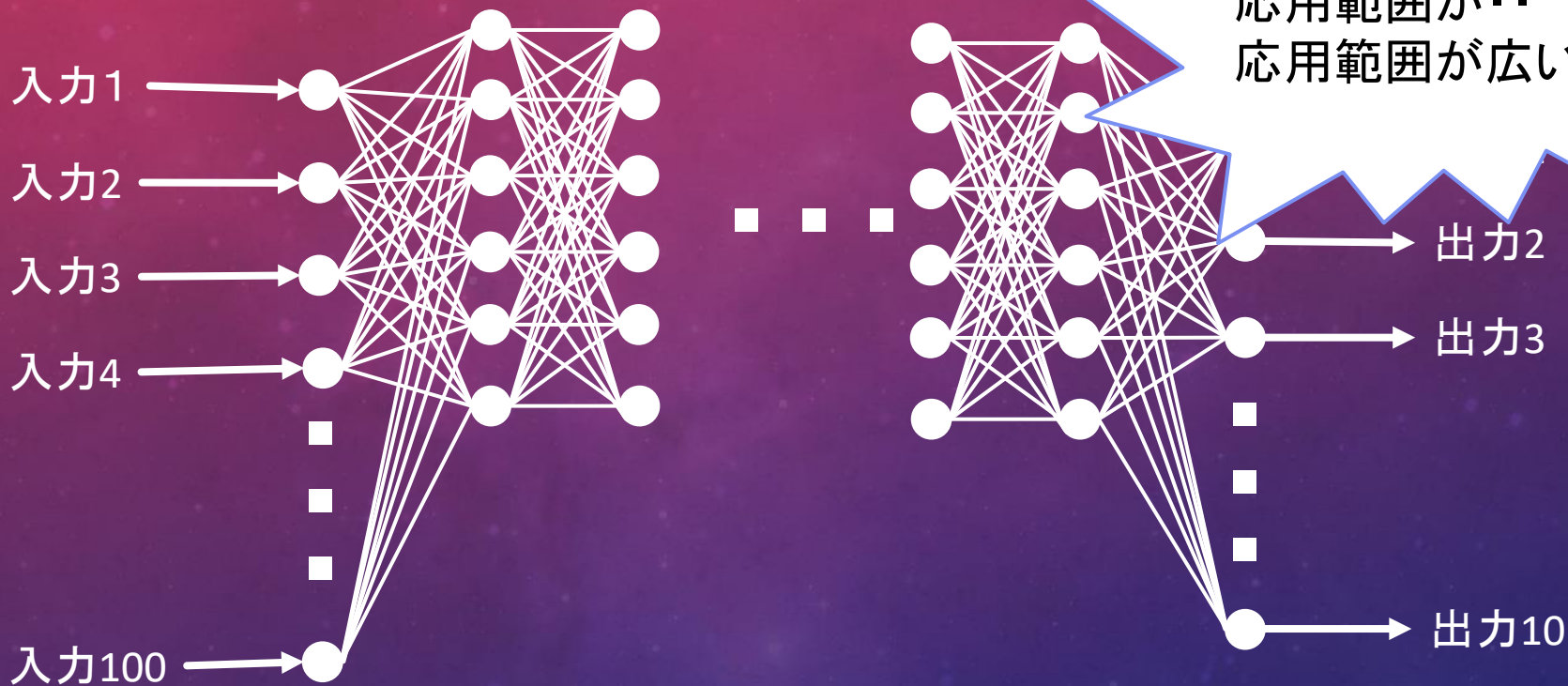
- 「入力」や「出力」「各層のノードの数」や「各層での処理」や「層の数」などを変更していけば様々な応用が可能！



様々な応用的な事を組み合わせたり、複数のモデルを組み合わせるなどしていくと stable diffusion や ChatGTP にもなっていきます。

# つまり？

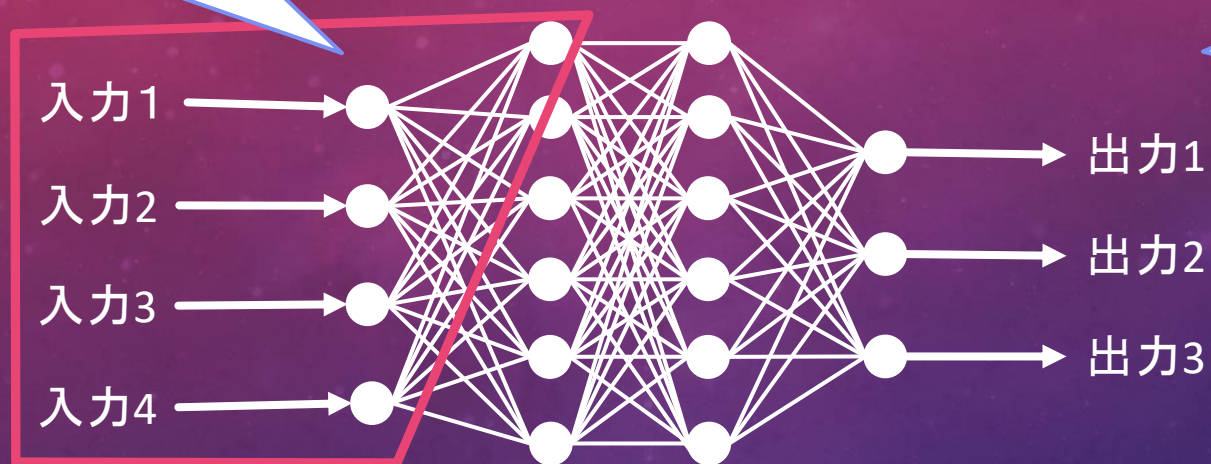
- 「入力」や「出力」「各層のノードの数」や「各層での処理」や「層の数」などを変更し



様々な応用的な事を組み合わせたり、複数のモデルを組み合わせるなどしていくと stable diffusion や ChatGTP にもなっていきます。

# 様々な応用できるが理解するのは大変なので簡単にする

1つのノードに着目する

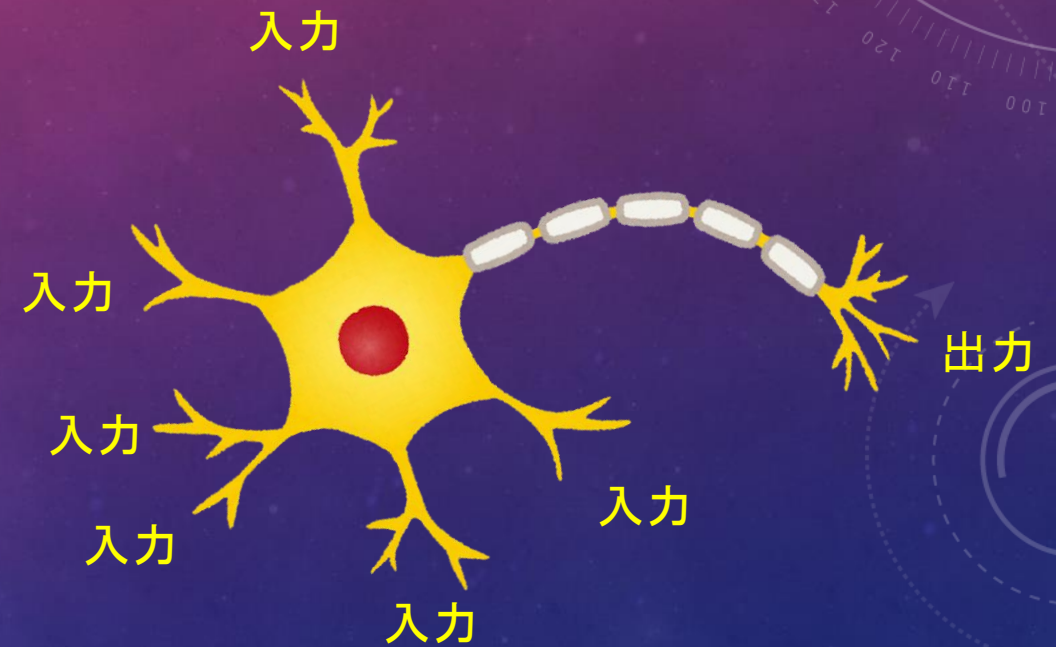
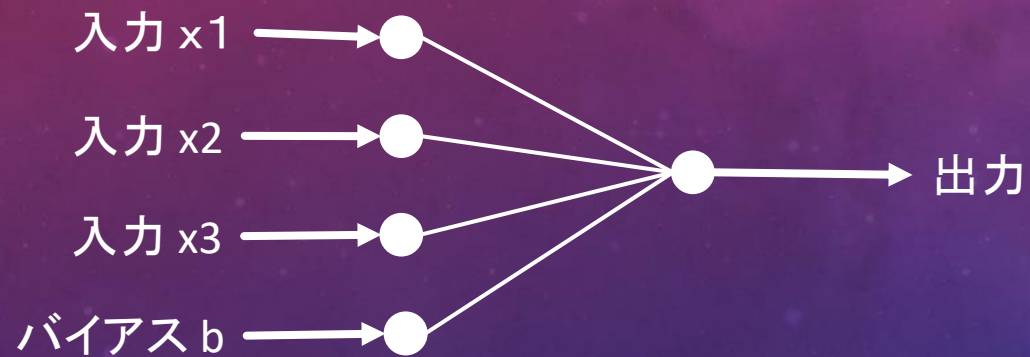


いきなりは無理っ！  
圧倒的無謀っ！

だから…  
簡単にするっ！

# 1つのノードにしたものがパーセプトロン

- パーセプトロンはディープではない単純な機械学習モデル
- 人間の脳内にある「ニューロン」と呼ばれる細胞を模したものとも呼ばれる。
- 人間の脳内にはこの「ニューロン」が10億個以上あります。

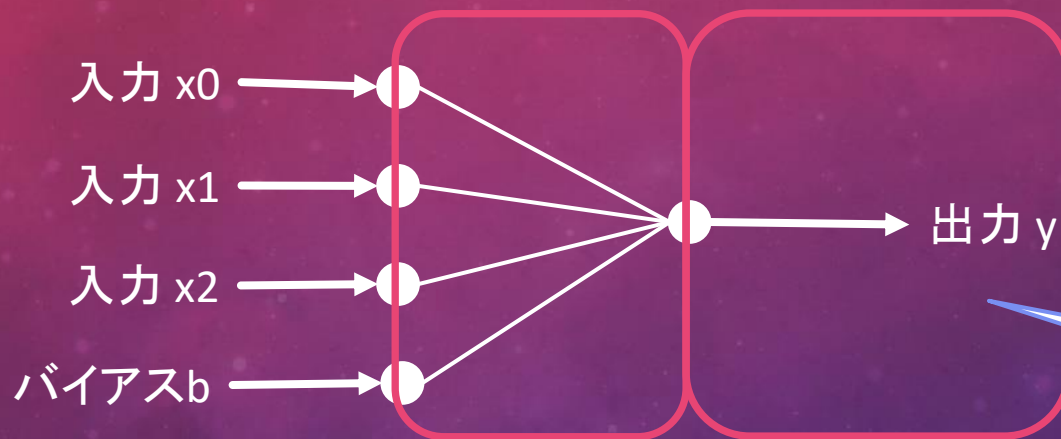


# パーセプトロン

ニューラルネットワークのノード1個分のお話

# パーセプトロンの仕組み

- それぞれの入力データに重みと呼ばれる係数をかけて全て足し合わせバイアスを足し、閾値を越えたら出力するもの。



- とすると

$$a = x_0 * w_0 + x_1 * w_1 + x_2 * w_2 + b$$

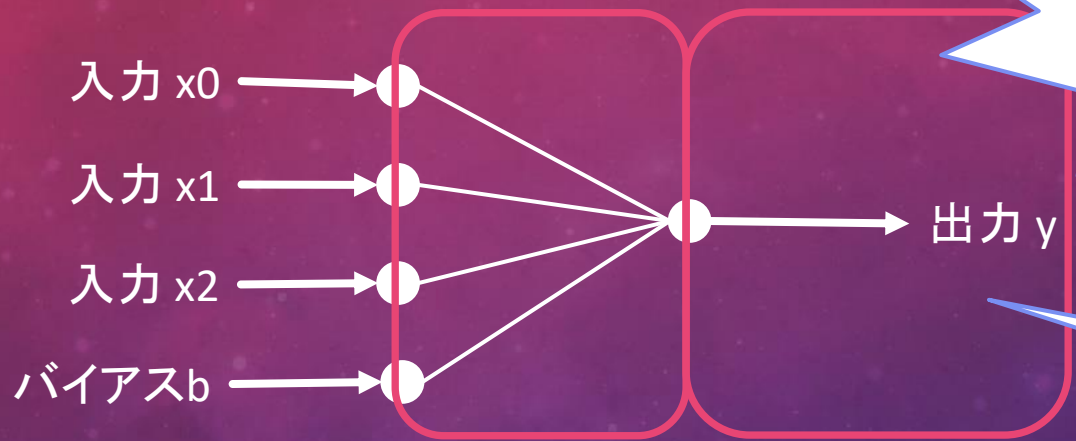
$$y = f(a)$$

としたものになります。(※wが学習するパラメータ)

これに対し、入力データと答えのサンプルを大量に用意して学習すれば機械学習が行えます。

# パーセプトロンの仕組み

- それぞれの入力データに重みと呼ばれる係数をかけて全て足し合わせバイアス



急に計算式なんか出されても  
わかるかっ！  
そんなもん！

- とすると

$$a = x_0 * w_0 + x_1 * w_1 + x_2 * w_2 + b$$
$$y = f(a)$$

ノードの前半で入力値と重みを  
掛けて足す

出てきた結果を閾値(0)を越えて  
いたら出力する。  
(活性化関数と呼びます)

としたものになります。(※wが学習するパラメータ)

これに対し、入力データと答えのサンプルを大量に用意して学習すれば機械学習が行えます。

# まだわかりづらいので入力データを1つにしてみる

- 入力データを1つにして活性化関数を省くと $y=x_0*w_0+b$ となるので



重みを「a」とすれば、ようやく

$$y=ax+b$$

が出てきた！



# まだわかりづらいので入力データを1つにしてみる

- 入力データを1つにして活性化関数を省くと $y=x_0*w_0+b$ となるので



やったぜ！きた！  
これで勝つる！

重みを「a」とすれば、ようやく

$$y=ax+b$$

が出てきた！

# とても分かりやすいモデルが出来たので機械学習してみる

- どういったタスクでやるか？
- 条件：入力データは1個

出力データも1個で活性化関数を使わないタスク = 数値予測の機械学習なら出来る。



$$y = ax + b$$

# とても分かりやすいモデルが出来たので機械学習してみる

- どういったタスクでやるか？
- 条件：入力データは1個

出力データも1個で活性化関数を使わないタスク = 数値予測の機械学習なら出来る。



$$y = ax + b$$

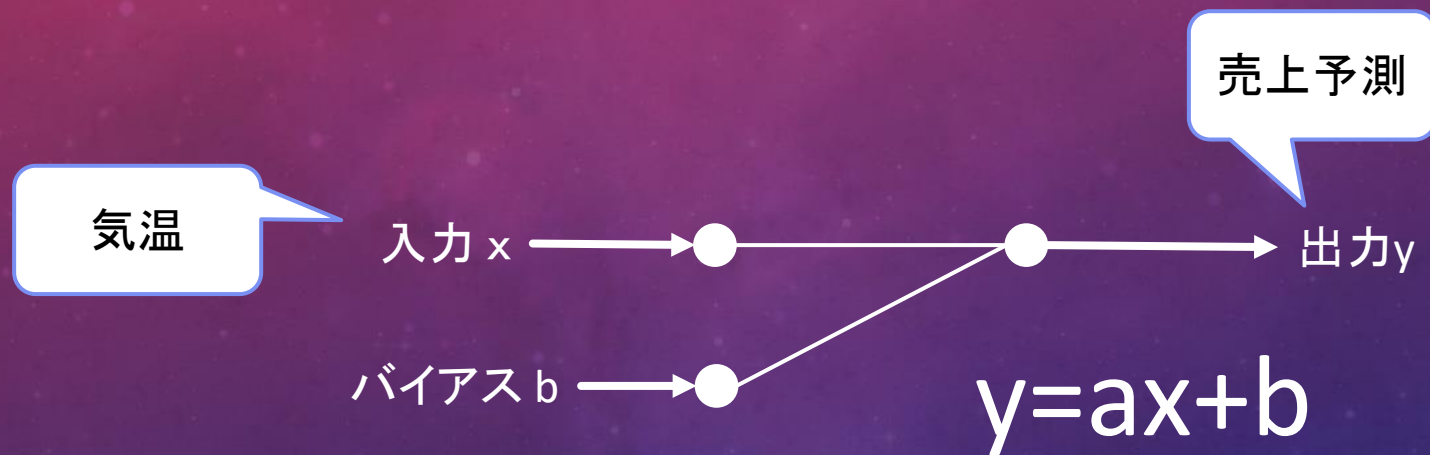
気温からアイスクリーム屋の売上を予測してみる。

# とても分かりやすいモデルが出来たので機械学習してみる

- どういったタスクでやるか？
- 条件：入力データは1個

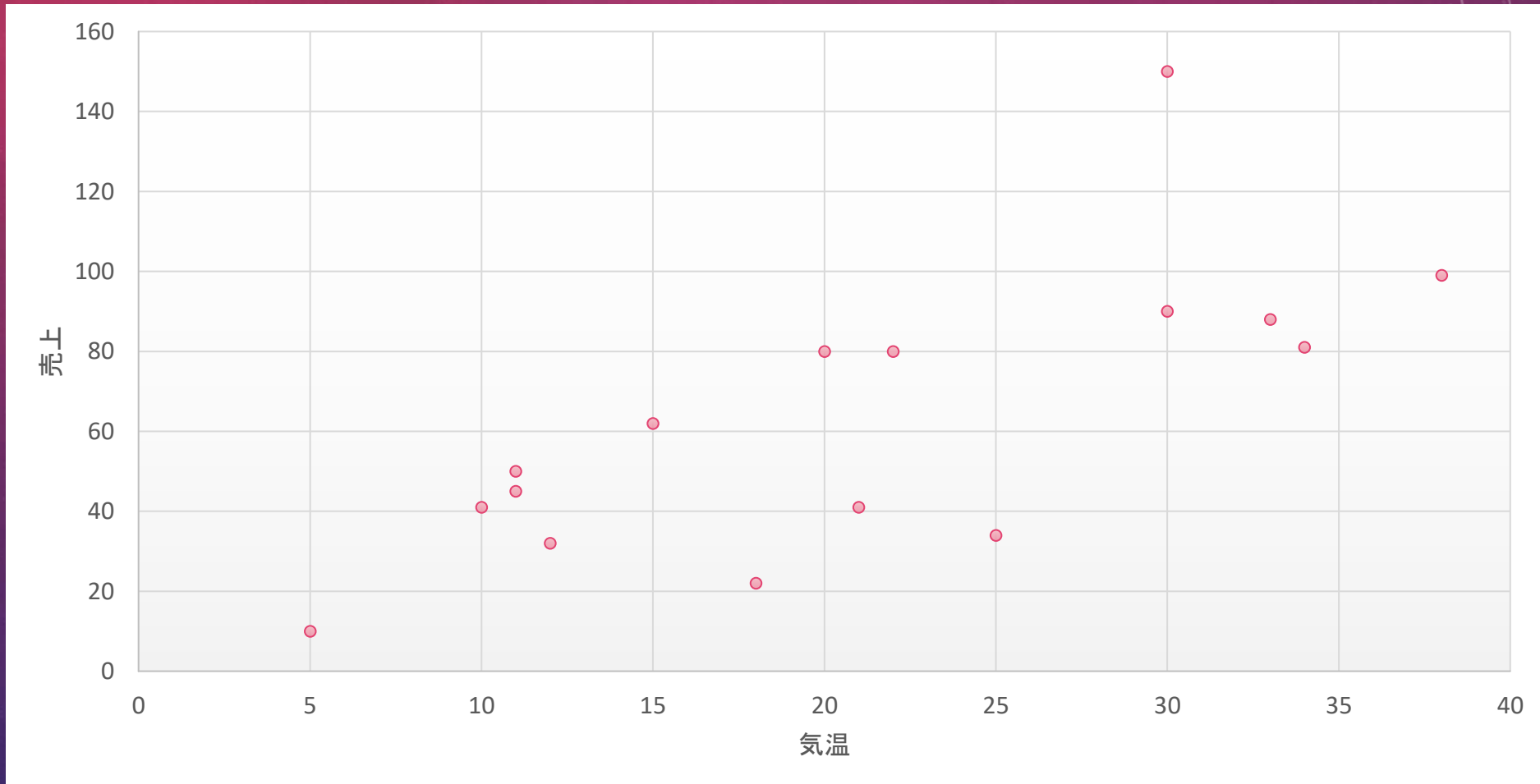
出力データも1個で活性化関数を使わないタスク = 数値予測の機械学習なら出来る。

- $y=ax+b$ で気温からアイスクリーム屋の売り上げ予測

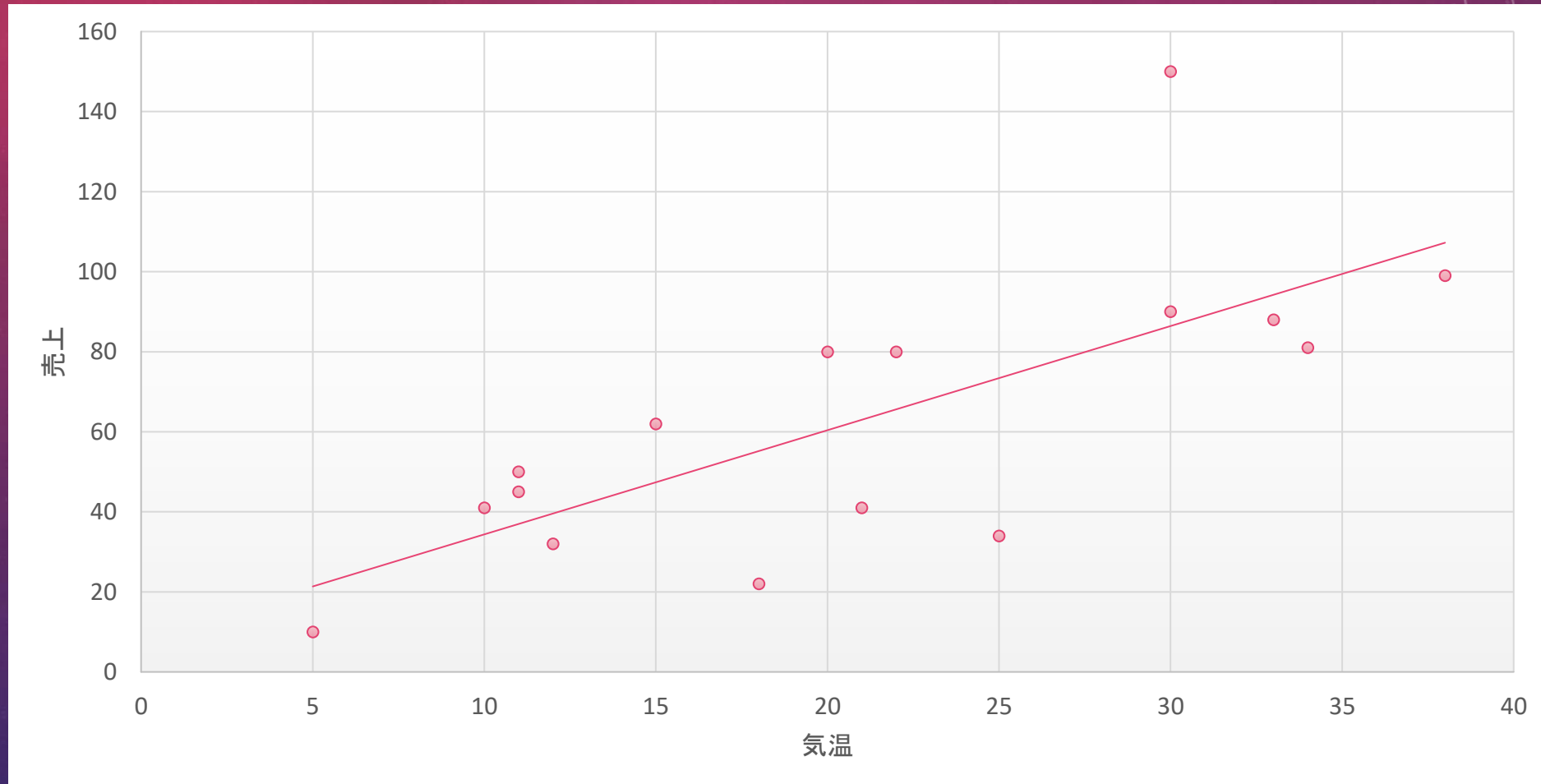


- として、適切な「a」と「b」の値が分かればある程度適切な予測が出来るはず！

# 仮にこういった気温に対する売上データがあったとする



赤線のような予測が出来るaとbの値が求めればOK  
 $y=ax+b$



# 機械学習を手動でやってみる

え！！手動で機械学習を！？  
出来らあっ！！

# 機械学習の学習手順

- ①入力データから機械学習アルゴリズムを使用して予測値の計算
  - ②予測値と正解の間の誤差を求めその誤差から学習パラメータに関する各微分値を計算
  - ③微分値から勾配法を用いてパラメータの更新
- 
- これを繰り返すのが機械学習の学習手順



# 機械学習の学習手順

- ①入力データから機械学習アルゴリズムを使用して予測値の計算
  - ②予測値と正解の間の誤差についての学習パラメータに関する微分値を計算
  - ③微分値から勾配法を用いてパラメータの更新
- これを繰り返すのが機械学習の学習手順

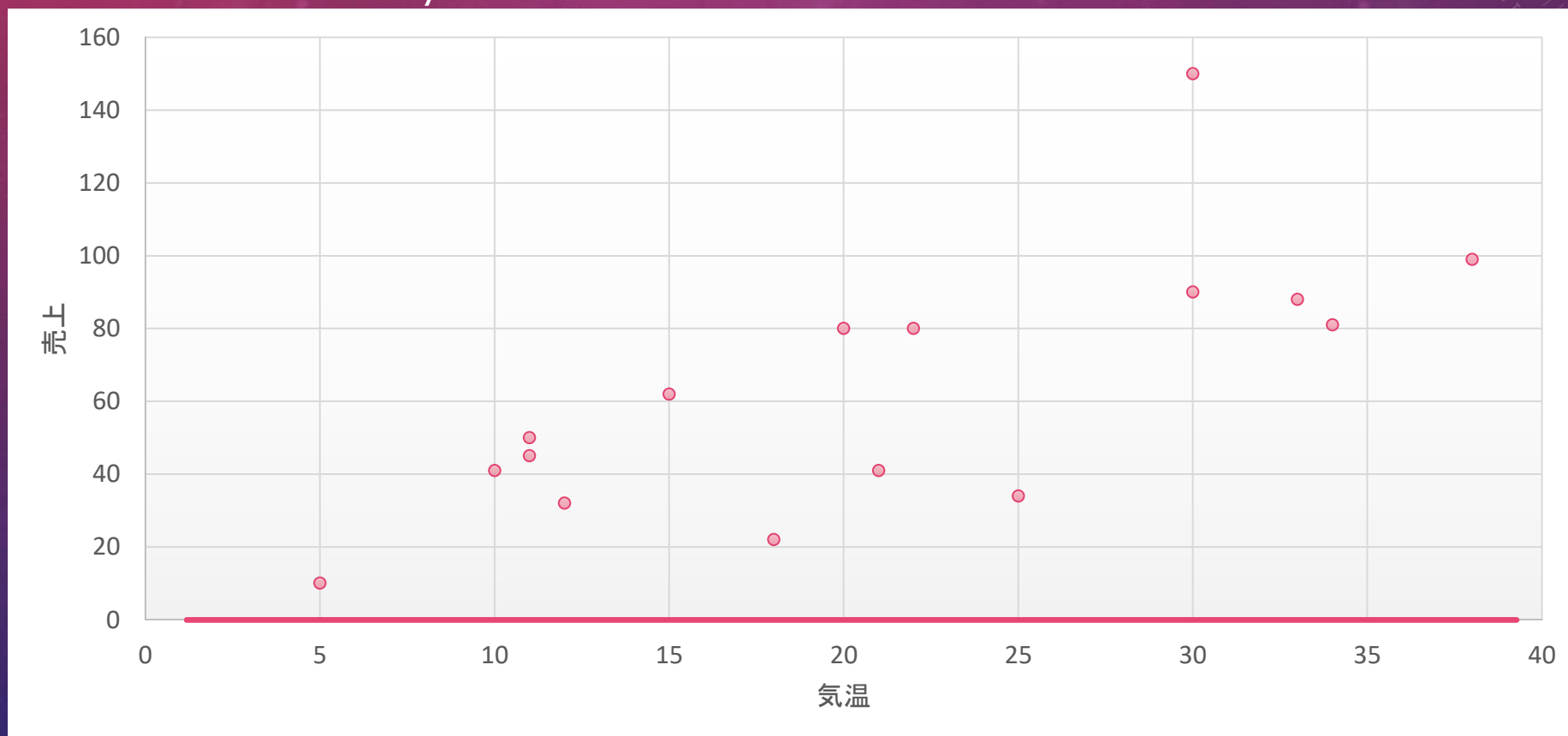
あーっ！いけません！  
分からない事出すの  
いけません！

あーっ！お客様！  
困ります！

# 分かりやすく手順説明

- $y=ax+b$ の $a$ と $b$ を適当に決める。(本来はランダムで決める)

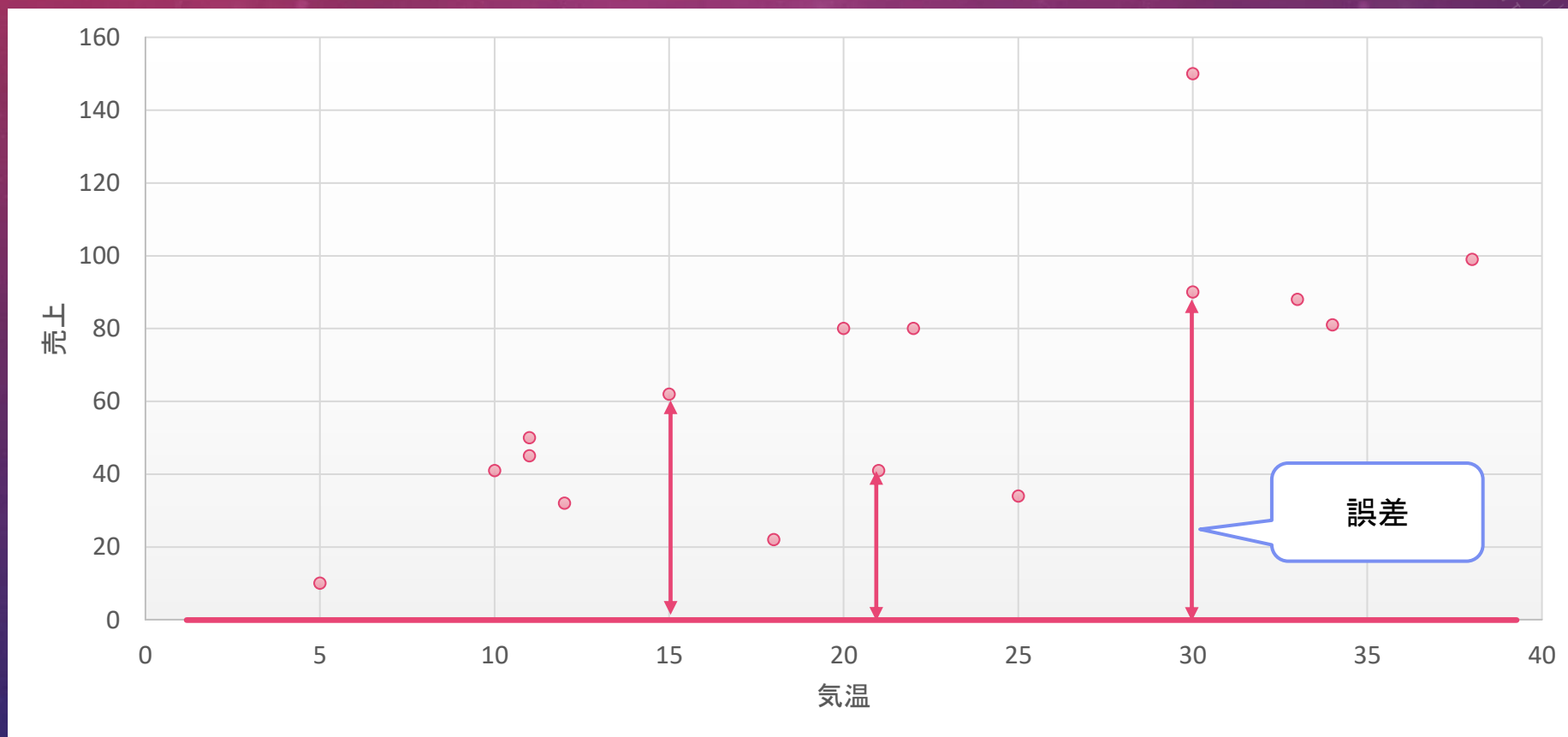
今回はわかりやすく「 $a=0$ 」「 $b=0$ 」として **$a$ のパラメータのみ**手動で機械学習します。 **$y=ax+b$**   
そうすると予測線は以下のように $y=0$ となる線となります。



# 分かりやすく手順説明

- そうすると $y=0$ の線と正解数値の間に誤差を計算することが出来るようになります。
- いくつかの点をランダムで取り誤差を計算します。

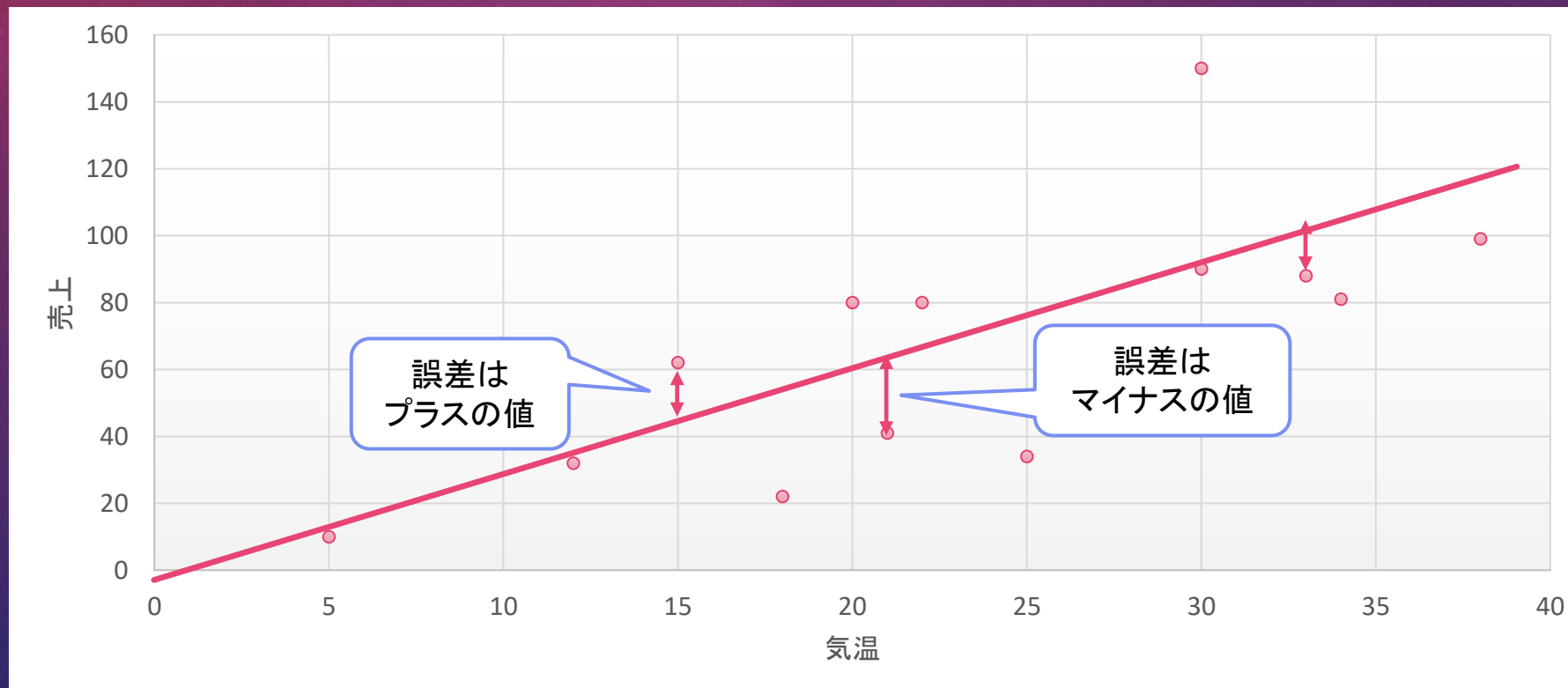
予測線 $y'=0$



# 注意点

- 誤差の求め方は 予測値: $y'$  正解数値: $y$
- とした場合、「 $y'-y$ 」としたいところですが、図のように学習が進むとプラスの誤差とマイナスの誤差があると計算がおかしくなるので、誤差を2乗したものを足し合わせるようにします。
- 計算式： $(y'-y)^2$

$$y' = ax + b$$



# 注意点（2乗和誤差）

- 誤差の求め方は 予測値: $y'$  正解数値: $y$
  - とした場合、「 $y'-y$ 」としたいところですが、図のように学習が進むとプラスの誤差とマイナスの誤差があると計算がおかしくなるので、誤差を2乗したものを足し合わせるようにします。
  - 計算式： $(y'-y)^2$
- この計算式を「**2乗和誤差**」と言います。

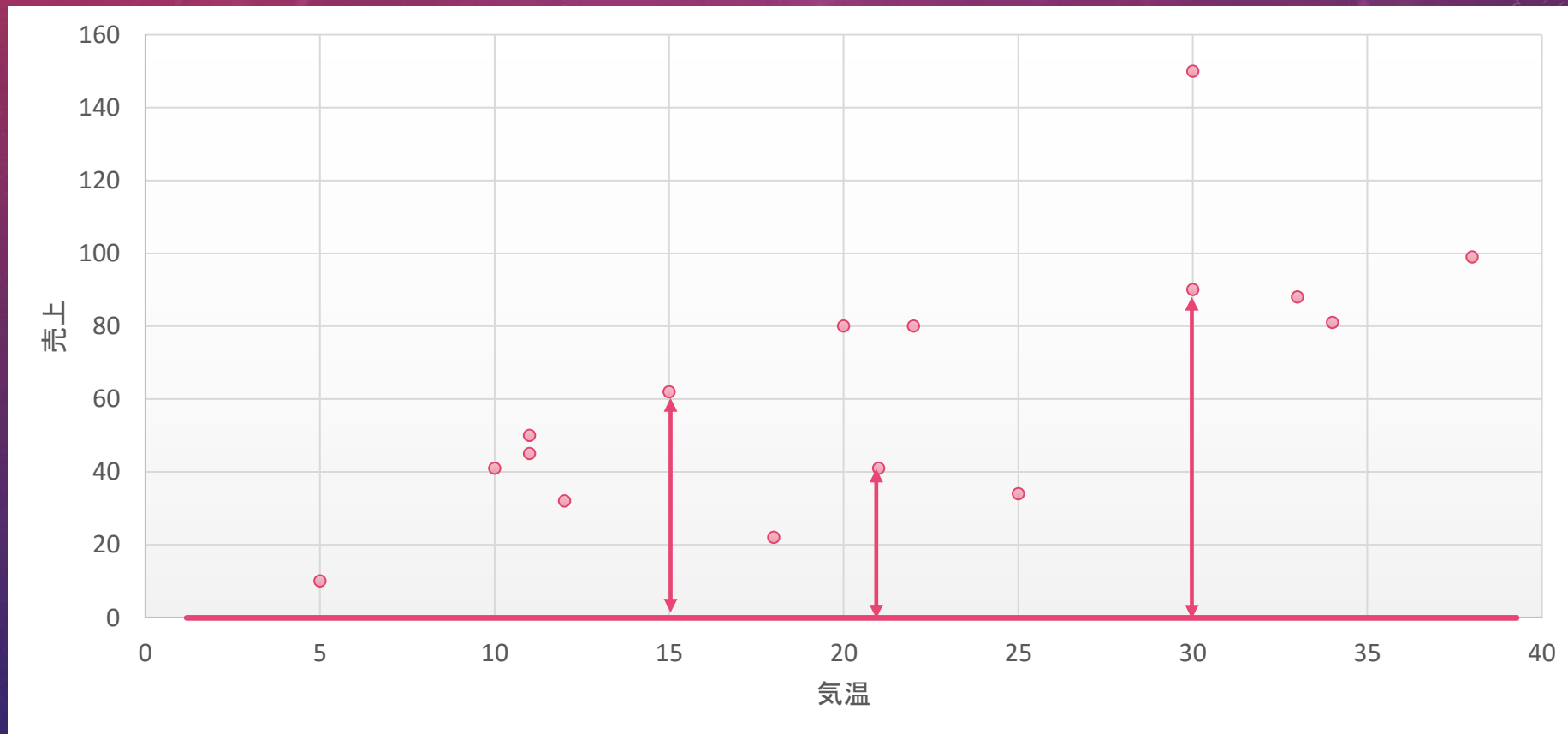
# いくつかのサンプルから2乗和誤差を求める。

- というわけで「 $a=0$ 」、「 $b=0$ 」でいくつかのデータをサンプリングして、2乗和誤差を求めます。

$(y'-y)^2$ をサンプル分合計するので

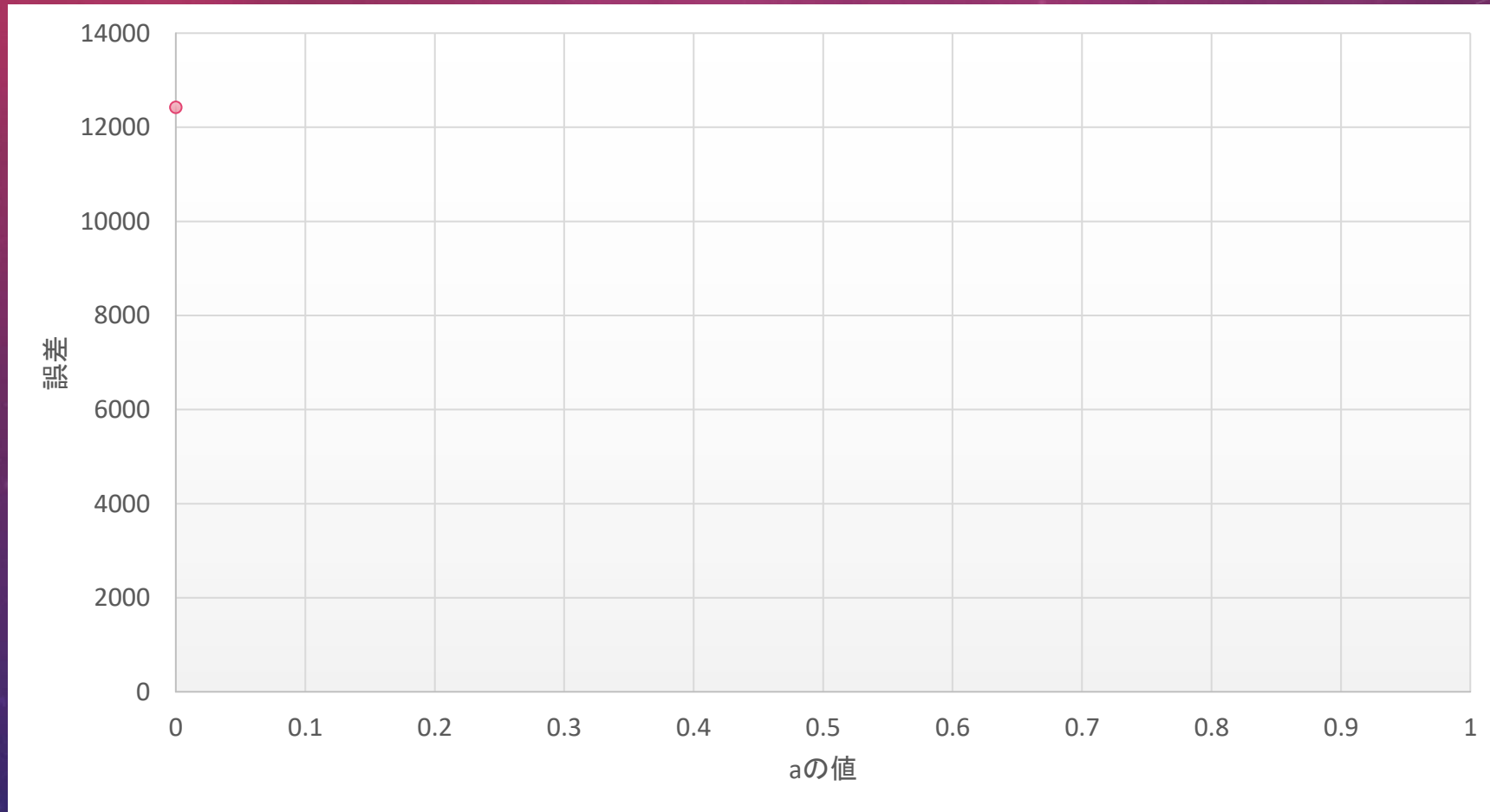
$60^2+40^2+85^2=12425$ が以下の3点の2乗和誤差

$y'=0$



# aの値と誤差のグラフを作ると

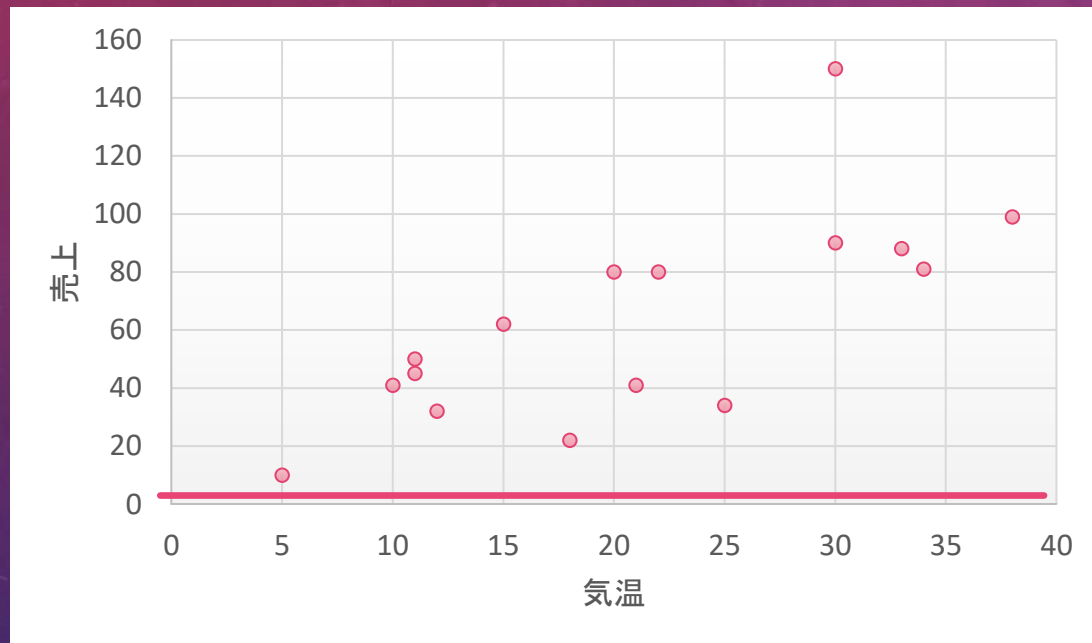
- こうなります。



# 2つのグラフを並べて考える

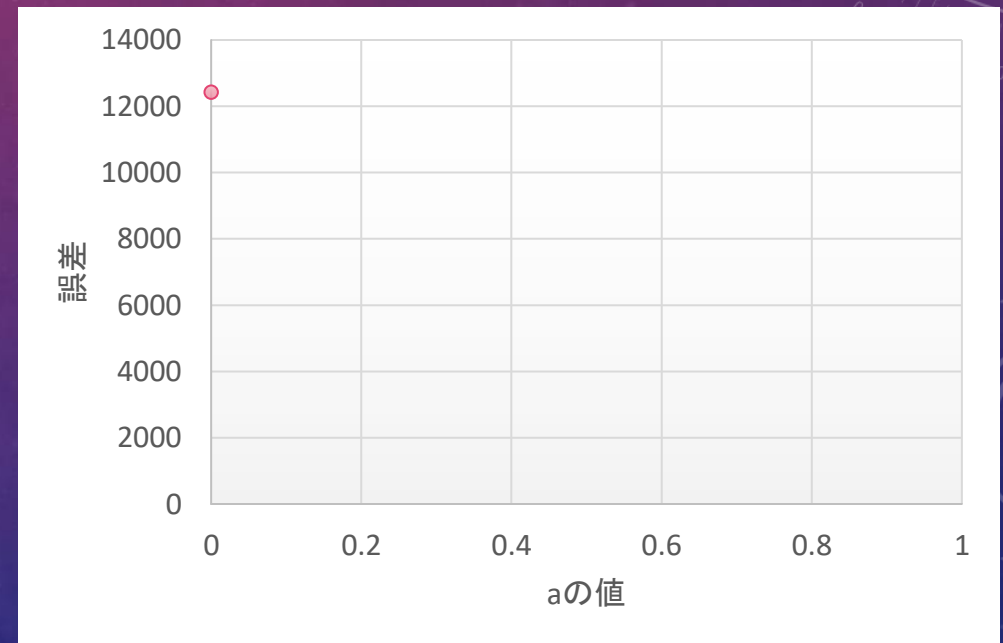
気温と売上の関係グラフ

$$y' = ax + b$$



誤差とaについてのグラフ

$$\text{誤差} = f(a)$$



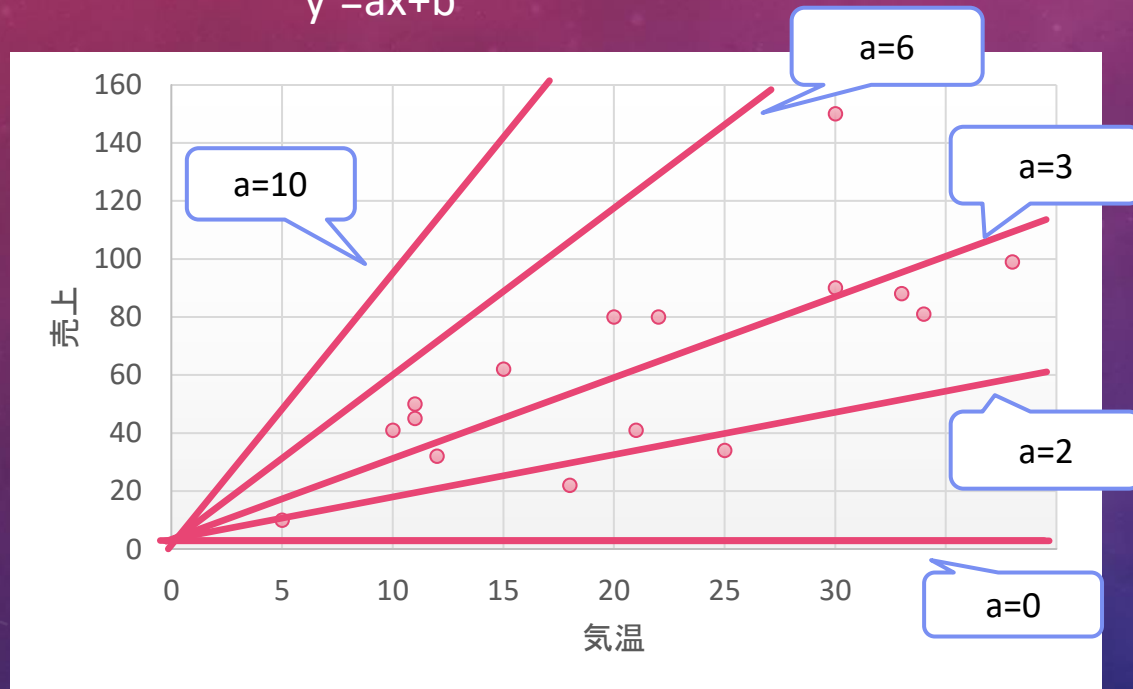


# 2つのグラフを並べて考える

- $a$ は気温と売上の関係グラフの傾きにあたるので $a$ の数値がプラスされていくと、予測線の傾きも大きくなります。
- 予測線の傾きが大きくなるほど恐らく誤差も小さくなり、ある1点を超えると誤差は再び大きくなると予想できます。

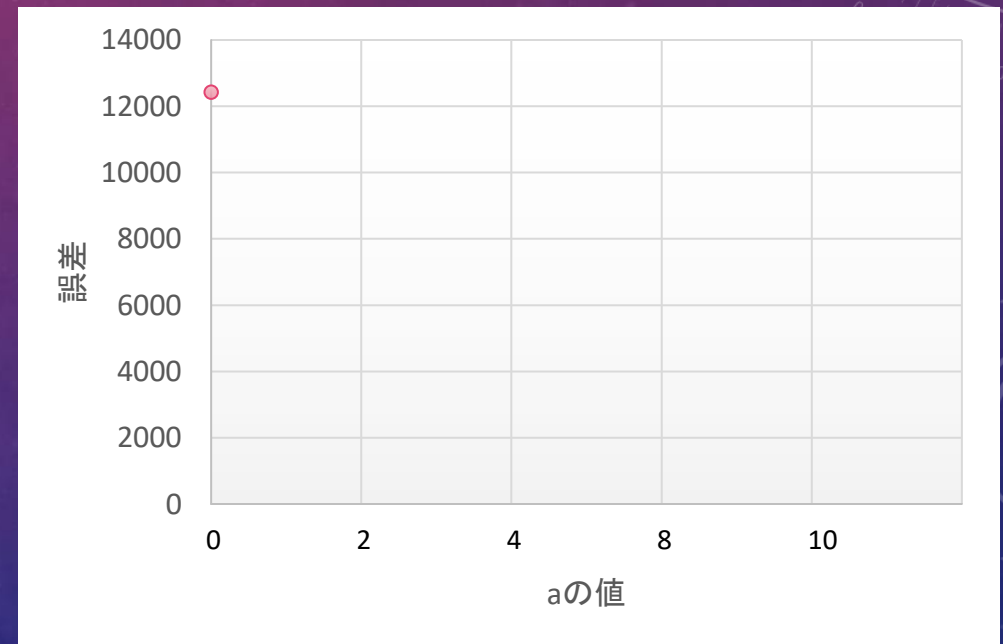
気温と売上の関係グラフ

$$y' = ax + b$$



誤差と $a$ についてのグラフ

$$\text{誤差} = f(a)$$

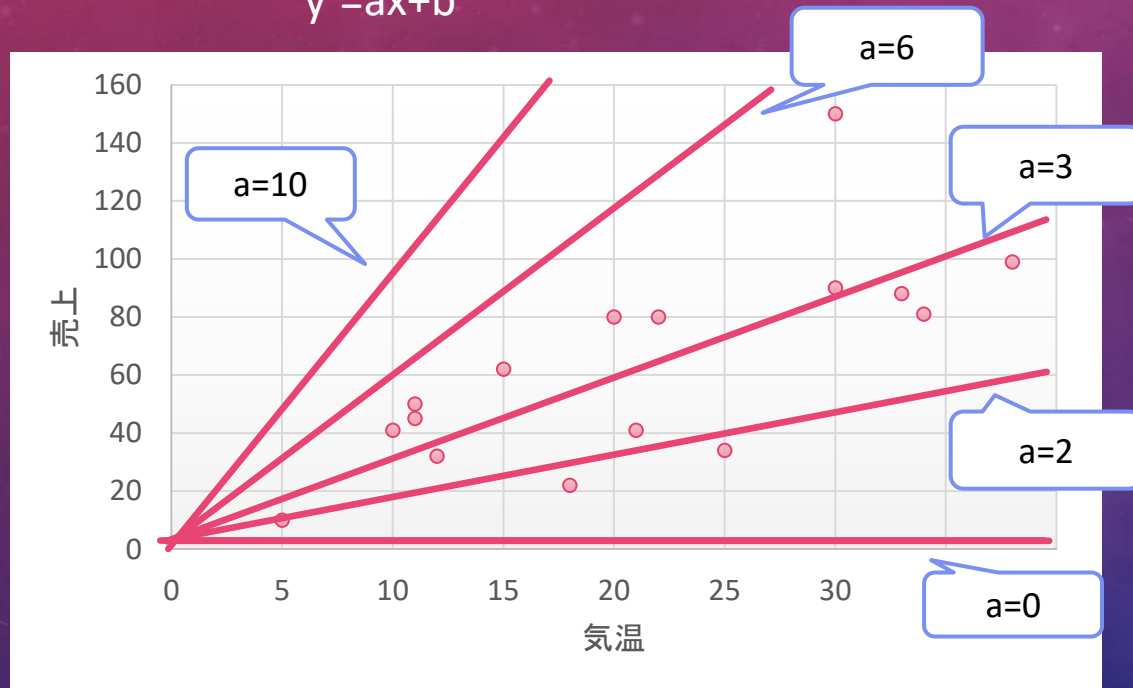


# 2つのグラフを並べて考える

- $a$ は気温と売上の関係グラフの傾きにあたるので $a$ の数値がプラスされていくと、予測線の傾きも大きくなります。
- 予測線の傾きが大きくなるほど恐らく誤差も小さくなり、ある1点を超えると誤差は再び大きくなると予想できます。

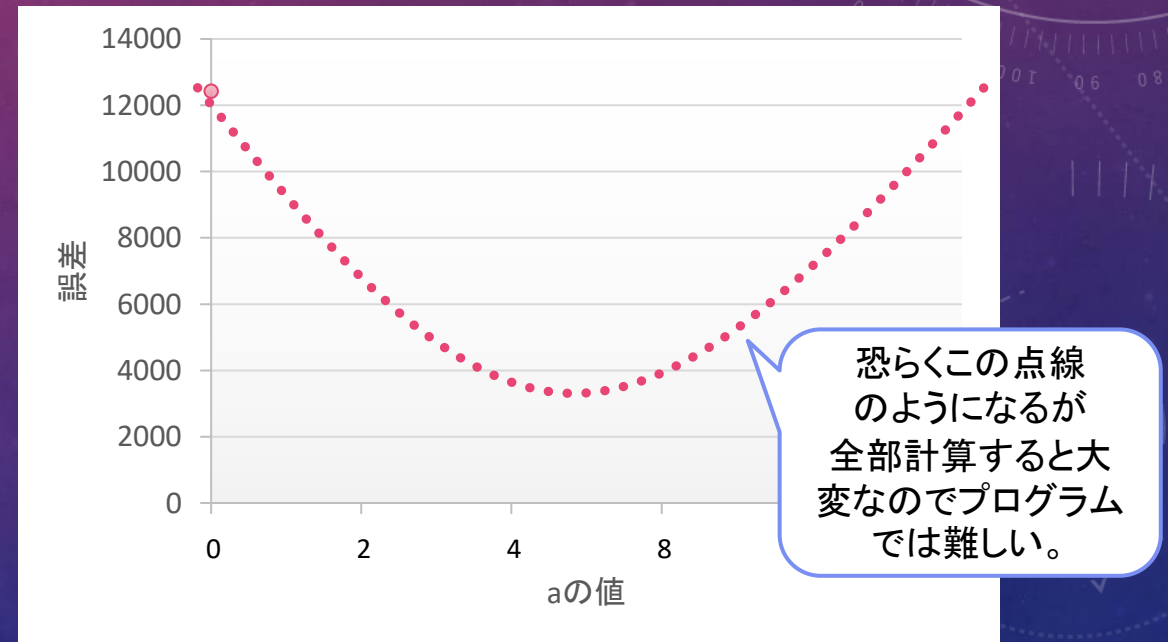
気温と売上の関係グラフ

$$y' = ax + b$$



誤差と $a$ についてのグラフ

$$\text{誤差} = f(a)$$

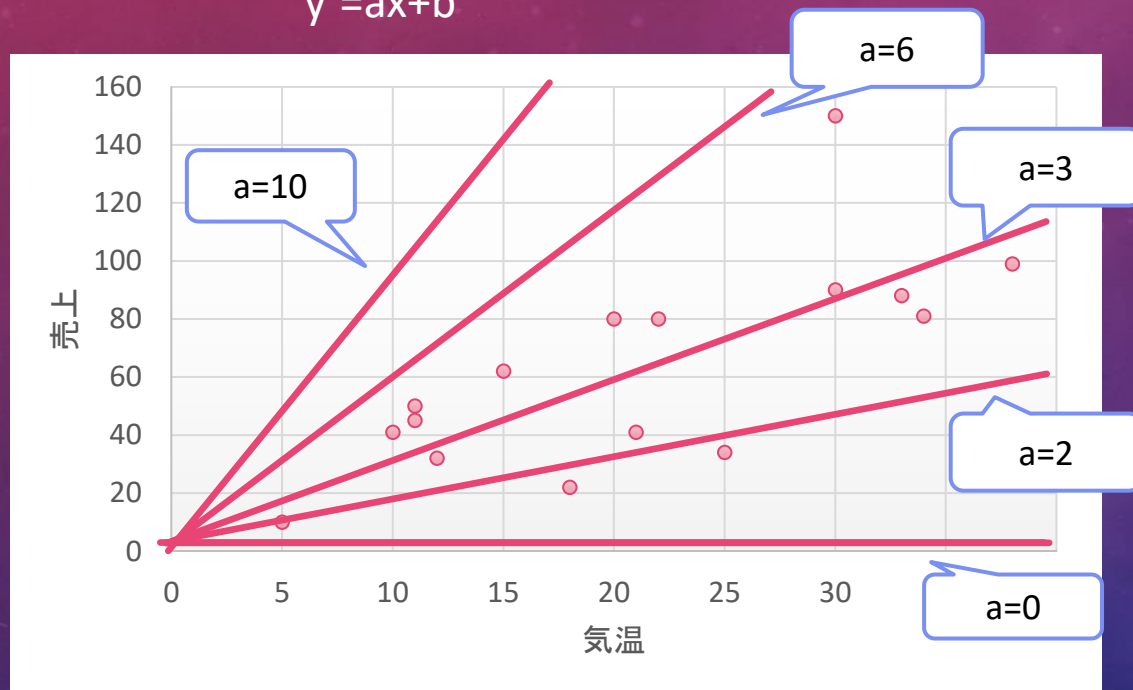


# 2つのグラフを並べて考える

- aが0のときはaをプラス方向にずらしたい。
- aが10ぐらいのときはaをマイナス方向にずらしたい。

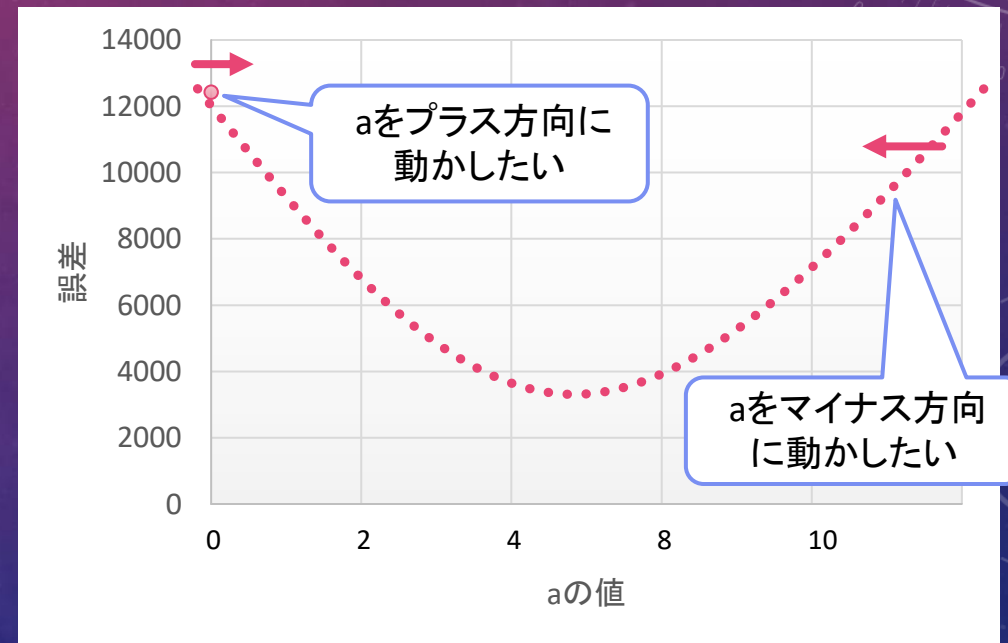
気温と売上の関係グラフ

$$y' = ax + b$$



誤差とaについてのグラフ

$$\text{誤差} = f(a)$$

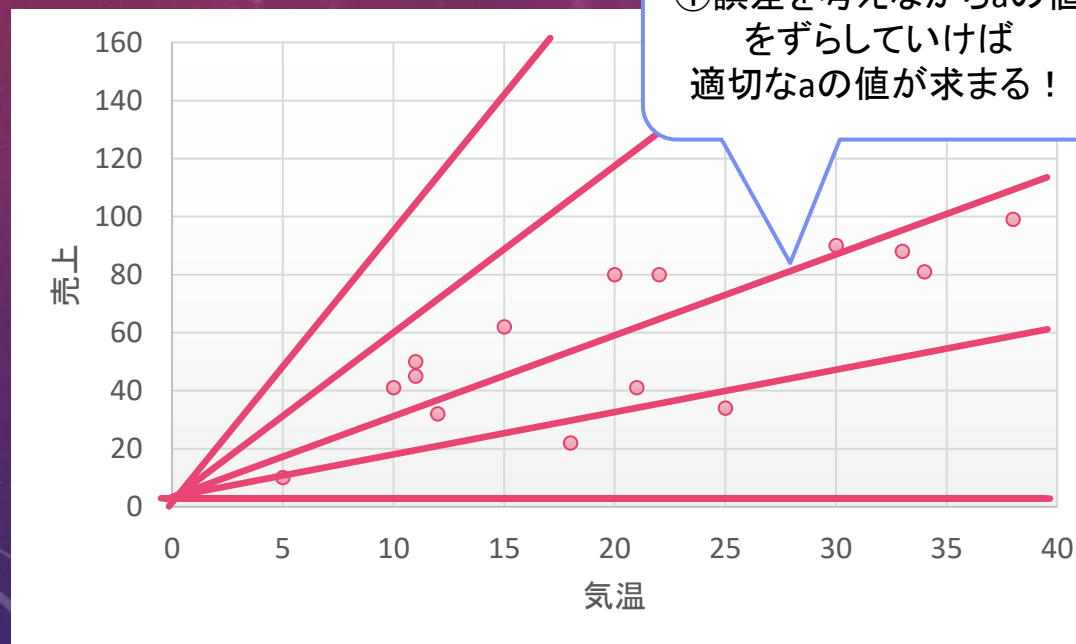


# 誤差とaについてのグラフで傾きを求める = 微分する

- 全部計算すると大変なので先ほど計算した誤差を使って誤差の傾きを求めればaを適切にずらしていく事ができる。
- $(y'-y)^2$ の微分値は $\frac{1}{2}(y' - y)$ となるのでこれが誤差とaについてのグラフの傾きなので
- **正解と予測値が分かっていたらその時のaをどちらに動かせばいいか計算できる！**

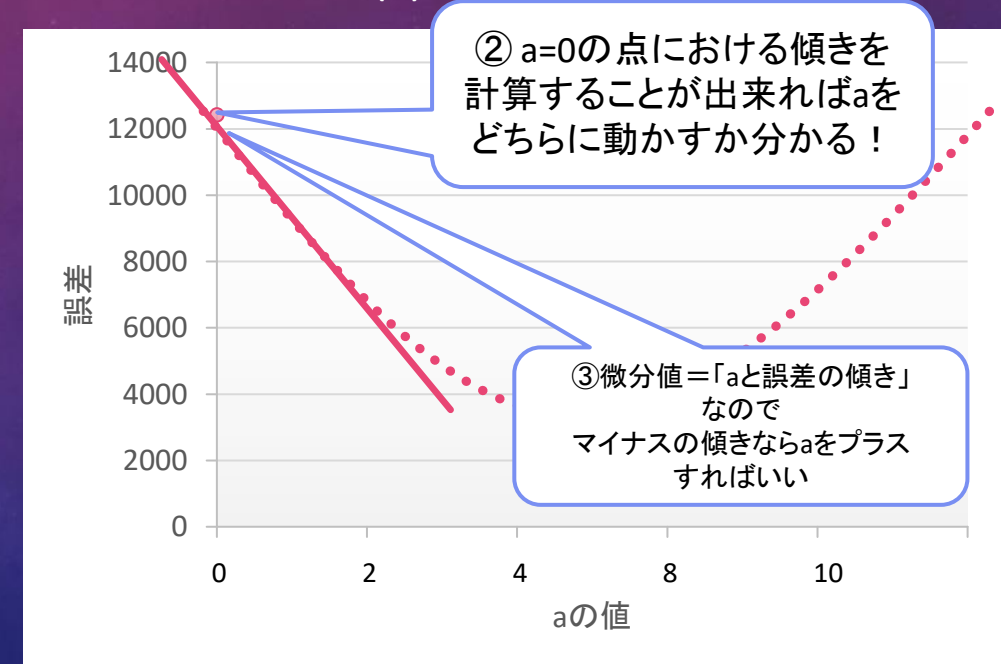
気温と売上の関係グラフ

$$y' = ax + b$$



誤差とaについてのグラフ

$$\text{誤差} = f(a)$$

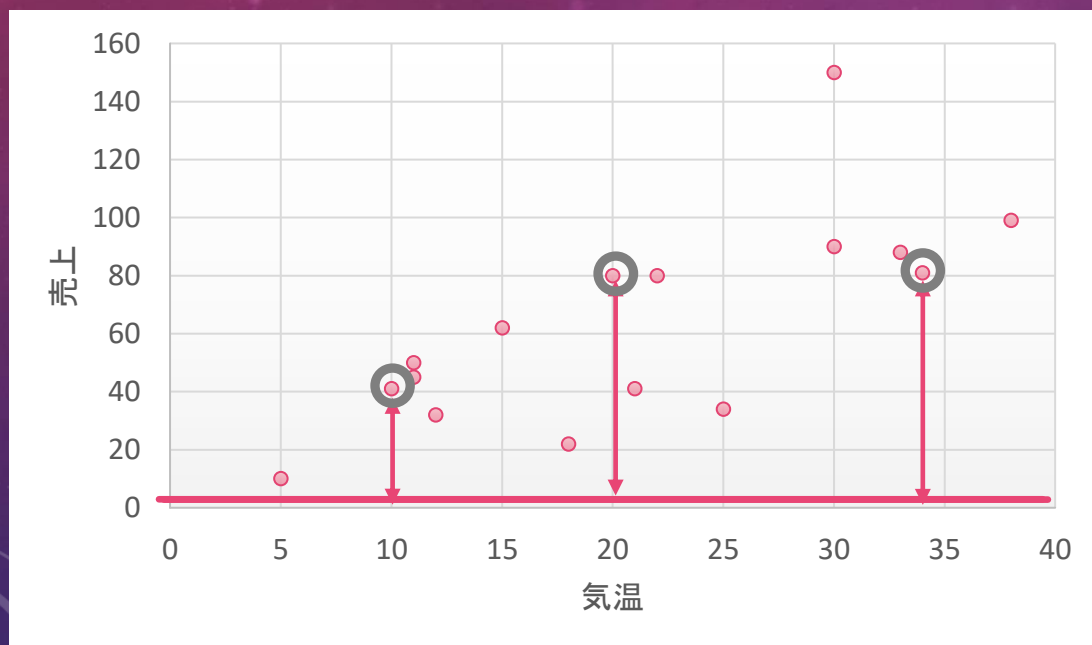


# 誤差の微分値からaの動かす数値を求める

- $\frac{1}{2}(y' - y)$ より図のランダムな3点の誤差の微分値の平均を求めると
- 左から-20、-40、-40となるので微分値の平均は-33.3...となります。
- これを**aの値の更新式**に代入して次に予測する際のaの値としていきます。

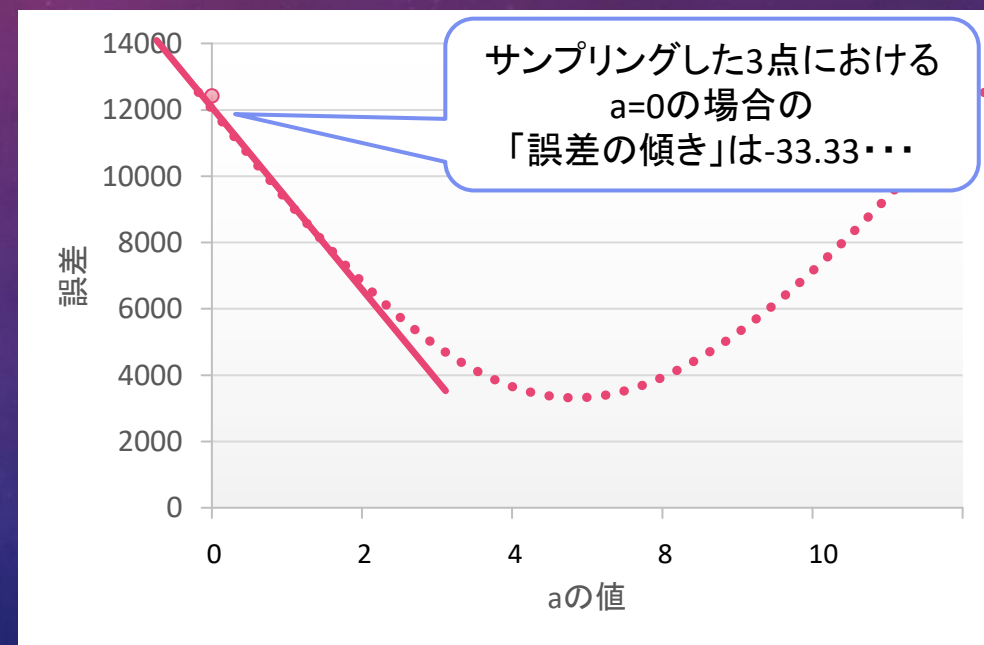
気温と売上の関係グラフ

$$y' = ax + b$$



誤差とaについてのグラフ

$$\text{誤差} = f(a)$$



# aの値の更新

一般的な勾配法では  
$$a \leftarrow a - \lambda \frac{\partial E}{\partial a}$$
と書かれてたりします。

- 今回は最も簡単な勾配法の計算式(次のaの値) $\leftarrow$ (今のaの値) $-\lambda * (\text{誤差の微分値})$ を使って計算します。

$\lambda$ (ラムダ)は人間が決めるハイパーパラメータでこの場合は0.1や0.01と言った値(1回にどれだけaを動かすかの基準)にします。

仮に $\lambda = 0.01$ とすると次のaの値は  $0 - 0.01 * (-33.3\dots)$

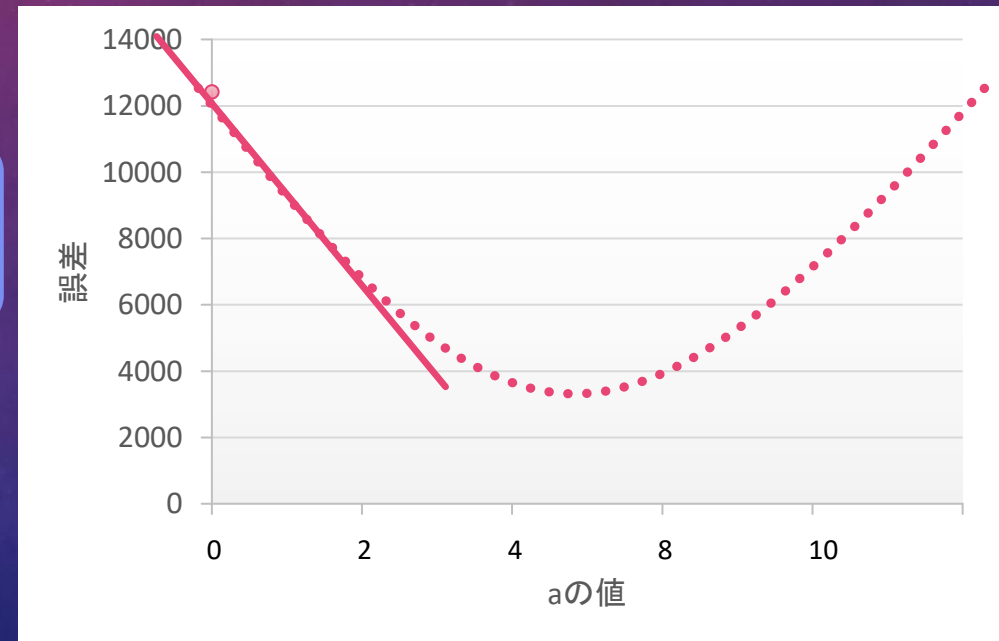
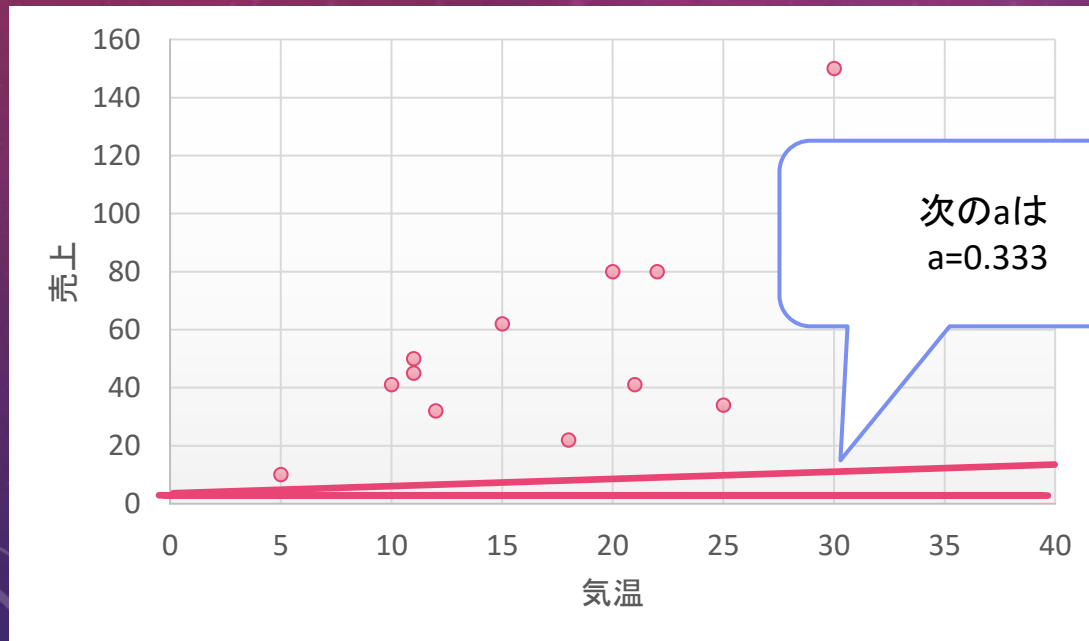
となるのでaの値は約0.333となります。

気温と売上の関係グラフ

$$y' = ax + b$$

誤差とaについてのグラフ

誤差 =  $f(a)$

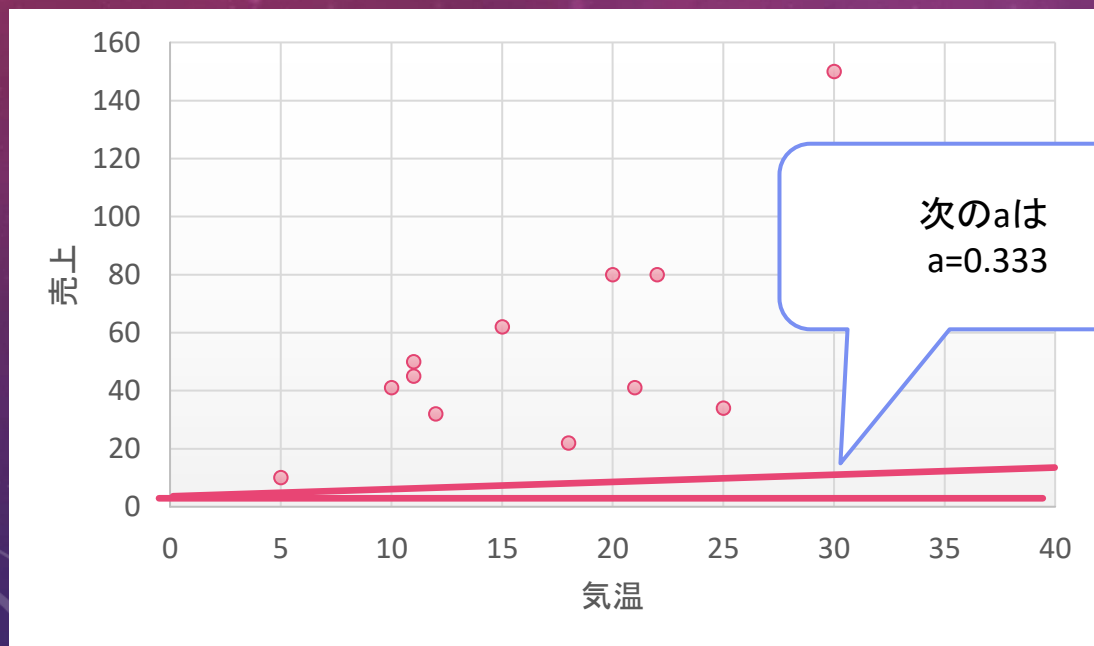


# 再び最初に戻り、aを何度も更新する

- ①  $y' = ax + b$  ( $a = 0.333$   $b = 0$ ) からランダムなサンプルを取り予測値を計算。
- ② 予測値と正解数値から微分値を計算。
- ③ 計算した微分値を使って次のaの値を求める。

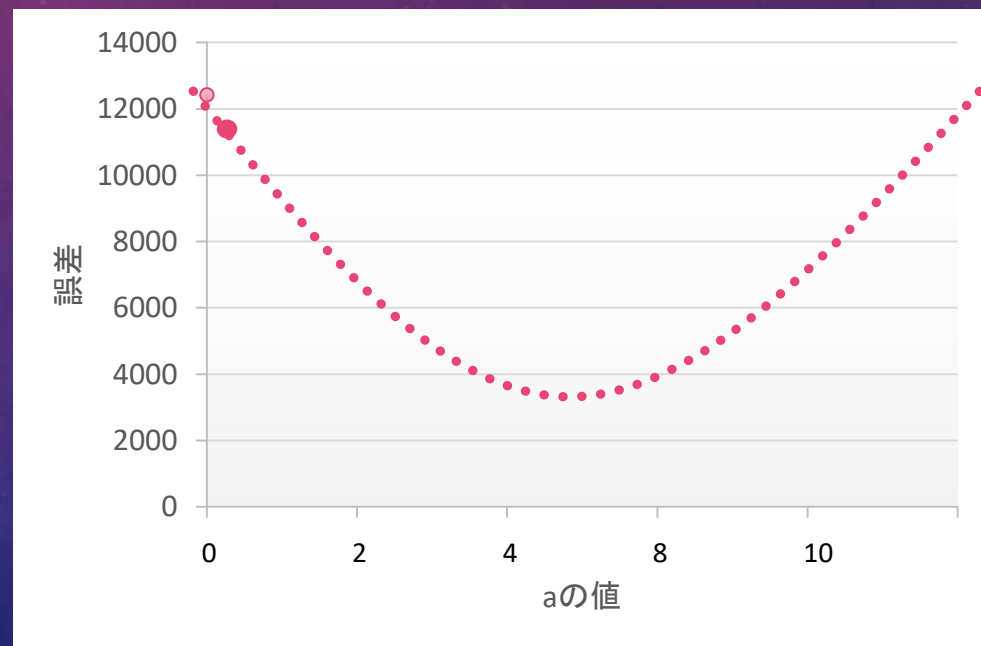
気温と売上の関係グラフ

$$y' = ax + b$$



誤差とaについてのグラフ

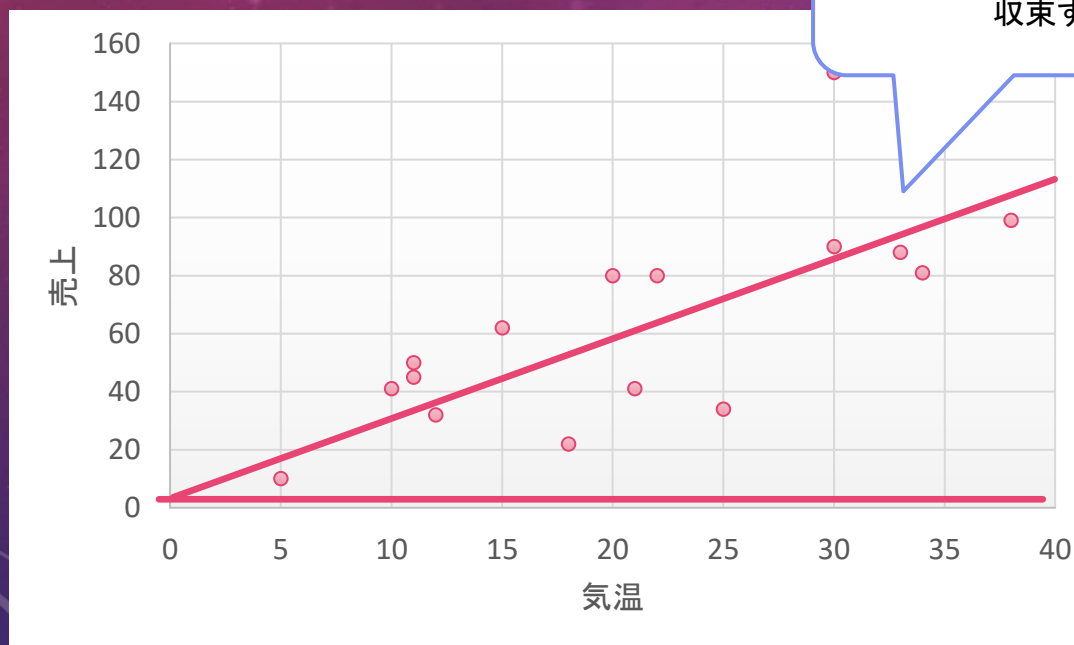
$$\text{誤差} = f(a)$$



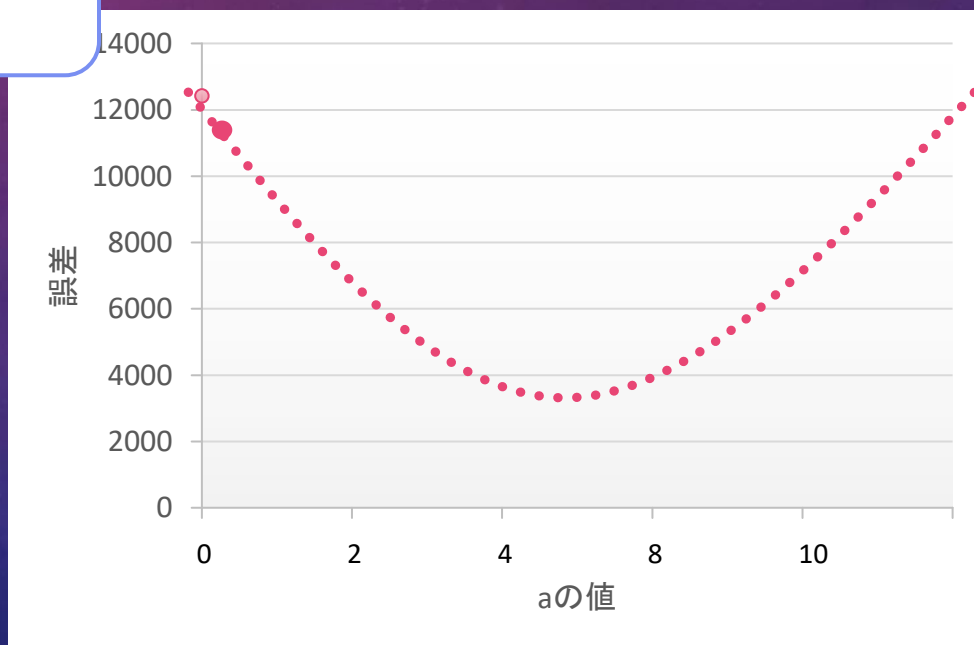
# 再び最初に戻り、aを何度も更新する

- ①  $y' = ax + b$  ( $a = 0.333$   $b = 0$ ) からランダムなサンプルを取り予測値を計算。
- ② 予測値と正解数値から微分値を計算。
- ③ 計算した微分値を使って次のaの値を求める。

気温と売上の関係グラフ  
 $y' = ax + b$



誤差とaについてのグラフ  
誤差 =  $f(a)$



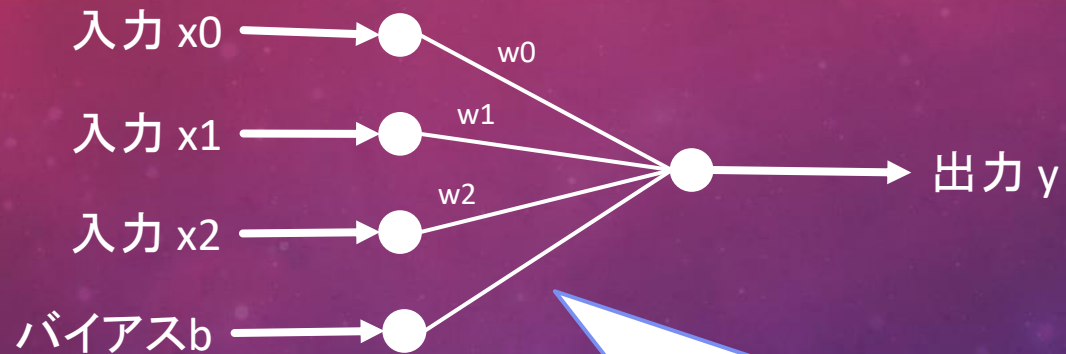


# 補足

- 今回は $y' = ax + b$ の $a$ のみ微分して求めていきましたが本来は「偏微分」という形で $a$ と $b$ 両方同時に更新していきます。
- ただ時間が無いので今日は省略！
- なお、今回のように $y = ax + b$ で1つの入力変数から予測数値を出す機械学習の事を「**単回帰分析**」と呼びます。
- 入力値が2つになれば $y = a_0 * x_0 + a_1 * x_1 + b$ となり、  
3つ以上になれば $y = a_0 * x_0 + a_1 * x_1 + a_2 * x_2 + \dots + b$ と変数が増えていくだけです。
- なお、この入力値が2変数以上で行う予測数値を出す機械学習の事は「**重回帰分析**」と呼んでいます。
- パーセプトロンの場合はこれに**活性化関数**を通して出力する形で2分類問題などに使われる機械学習となります。

# 改めてパーセプトロンの場合

- それぞれの入力データに重みと呼ばれる係数をかけて全て足し合わせバイアスを足し、閾値を越えたら出力するもの。



線1本ごとに更新パラメータがある

- とすると

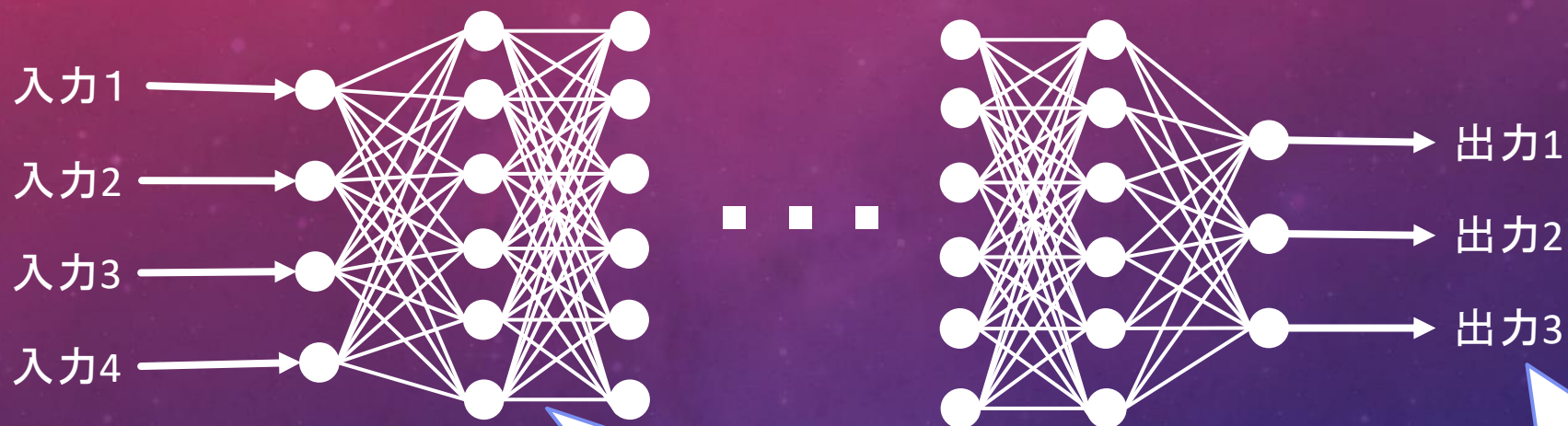
$$a = x_0 * w_0 + x_1 * w_1 + x_2 * w_2 + b$$

$$y = f(a)$$

としたものになります。(※wとbが学習するパラメータ)

# 改めてニューラルネットワークの場合

- ーの部分1本1本にそれぞれ重み $w$ が存在し、各層にバイアスがあるので全部偏微分して
- 微分値からパラメータを更新しています。
- この結果複雑な人間のような判断もできるようになる！



線1本ごとに更新パラメータがある

やはり計算量...!!  
計算量こそすべてを  
解決する...!!

# 改めて機械学習の学習手順

- ①入力データから機械学習アルゴリズムを使用して予測値の計算
  - ②予測値と正解の間の誤差についての学習パラメータに関する各微分値を計算
  - ③微分値から勾配法を用いてパラメータの更新
- 
- どのようなモデルで有ろうとも、これを繰り返すのが機械学習の学習手順
- 
- つまり

$y=ax+b$ の機械学習がニューラルネットワークの基本

The background features a gradient from purple to blue with bokeh effects. On the left, there are technical diagrams including a large circular scale with numerical markings (40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) and various circular arrows and lines.

ありがとうございます

第1回  $y=ax+b$ から始める初心者向けML講座  
次回は活性化関数とニューラルネットワーク（多分）