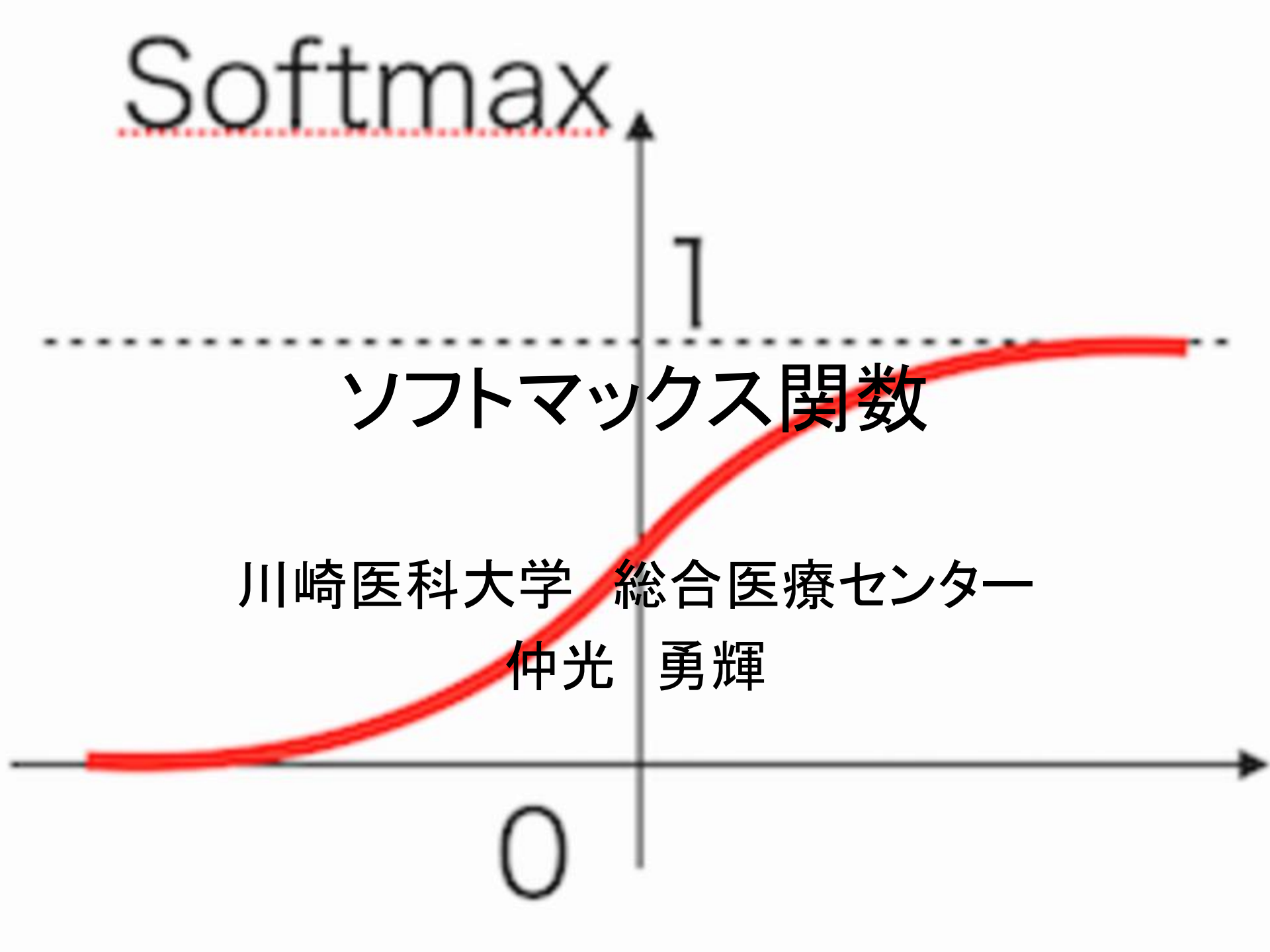
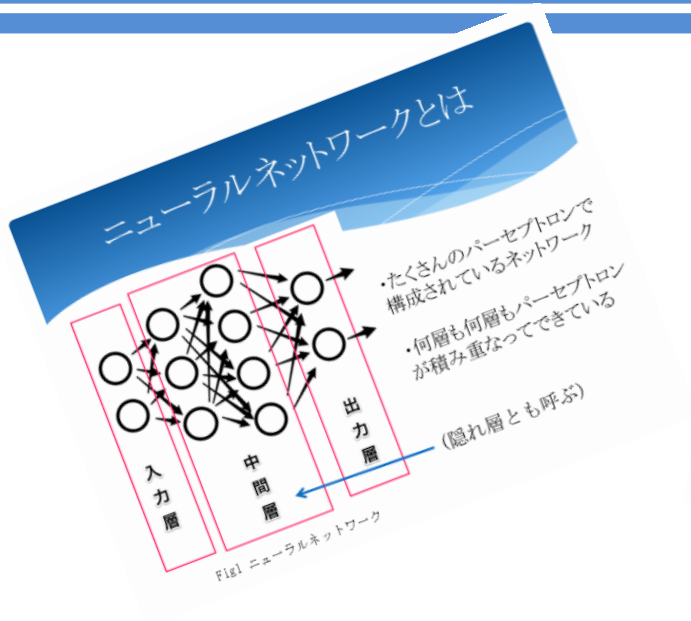
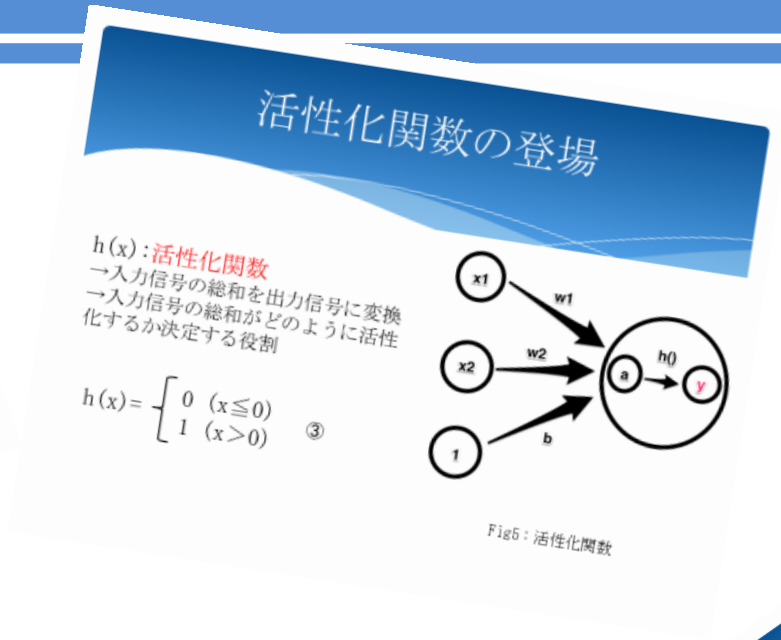


ソフトマックス関数

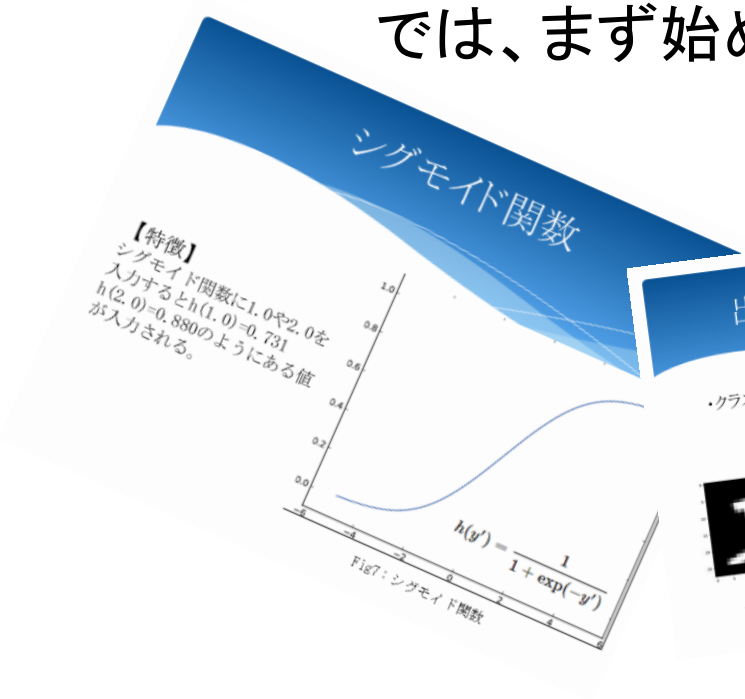
川崎医科大学　総合医療センター
仲光　勇輝

今回も実装を省略させていただきます
ご了承ください。

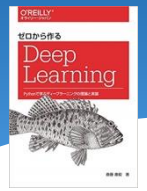




では、まず始めに前回の続きから…



3章でのキーワード



p.82に記載

・ニューラルネットワークでは、活性化関数としてシグモイド関数やReLU関数のような滑らかに変化する関数を利用する。

・NumPyの多次元配列をうまく使うことで、ニューラルネットワークを効率よく実装することができる。

・機械学習の問題は、回帰問題と分類問題に大別できる。

・出力層で使用する活性化関数は、回帰問題では恒等関数、分類問題ではソフトマックス関数を一般的に利用する。

・分類問題では、出力層のニューロンの数を分類するクラス数に設定する。

・入力データのまとまりをバッチと言い、バッチ単位で推論処理を行うことで、計算を高速に行うことができる。

出力層の設計

- ニューラルネットワークは、分類問題と回帰問題の両方に用いることが可能。
- どちらの問題を用いるかで出力層の活性化関数を変更する必要がある。
- 回帰問題では恒等関数を、分類問題ではソフトマックス関数を使用する。

出力層の設計

- 回帰問題

ある入力データから、(連続的な)数値の予測を行う問題

→たとえば、猫の画像を見せてその画像から猫の身長や体重を予測するような問題

- 分類問題

データがどのクラスに属するか、という問題

→例えば、猫の画像を見せたときにそれは猫か？犬か？を分類するような問題

回帰問題　分類問題

過去のデータ

分類

新しいレストランが
好き or 好きじゃない
か予測

回帰

新しいレストランに
今後何回訪れるか予測

出力層の設計

➢分類問題

- ✓ データのクラス分け
- ✓ ソフトマックス関数

➢回帰問題

- ✓ ある入力から何かしらの数値を予測するもの
- ✓ 恒等関数

恒等関数

- 受け取ったそのままの値を変換せずに出力
- 回帰問題で使用

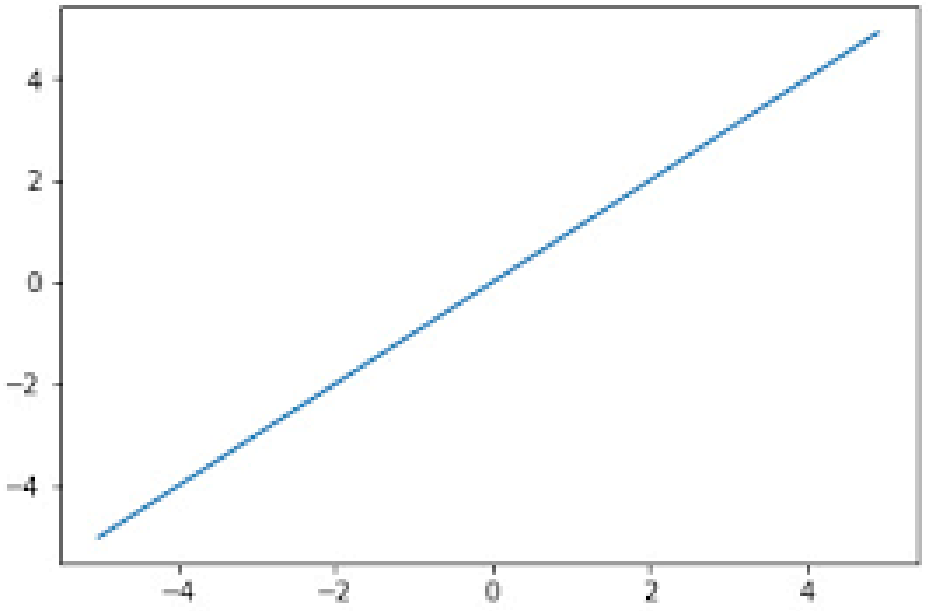


Fig12:恒等関数

ソフトマックス関数

- 分子は入力信号akの指数関数
  分母は全ての入力信号の指数関数の和から構成
- n個ある出力層のk番目の出力ykを求める式である.
- 出力の各ニューロンがすべての入力信号から影響を受ける.
- 分類問題で使用

$$y_k = \frac{exp(a_k)}{\sum_{i=1}^{n} exp(a_i)}$$

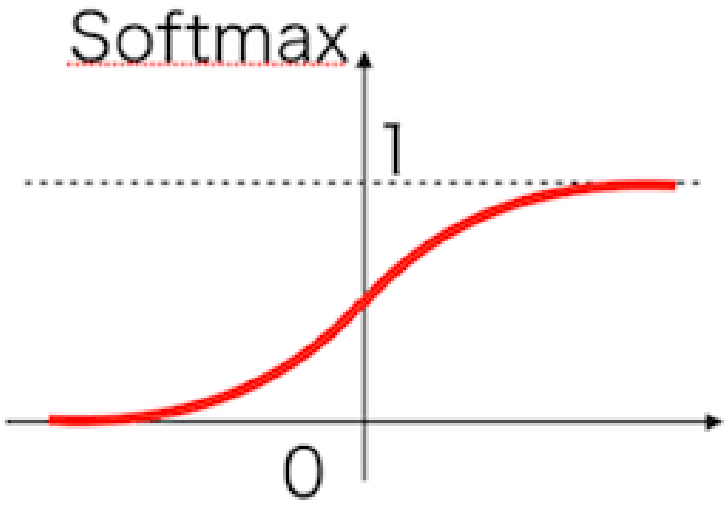
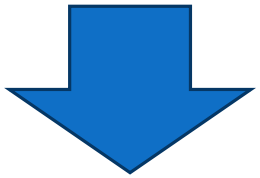


Fig13:ソフトマックス関数

ソフトマックス関数の特徴

・ソフトマックス関数の出力は0～1.0
・出力の総和は1

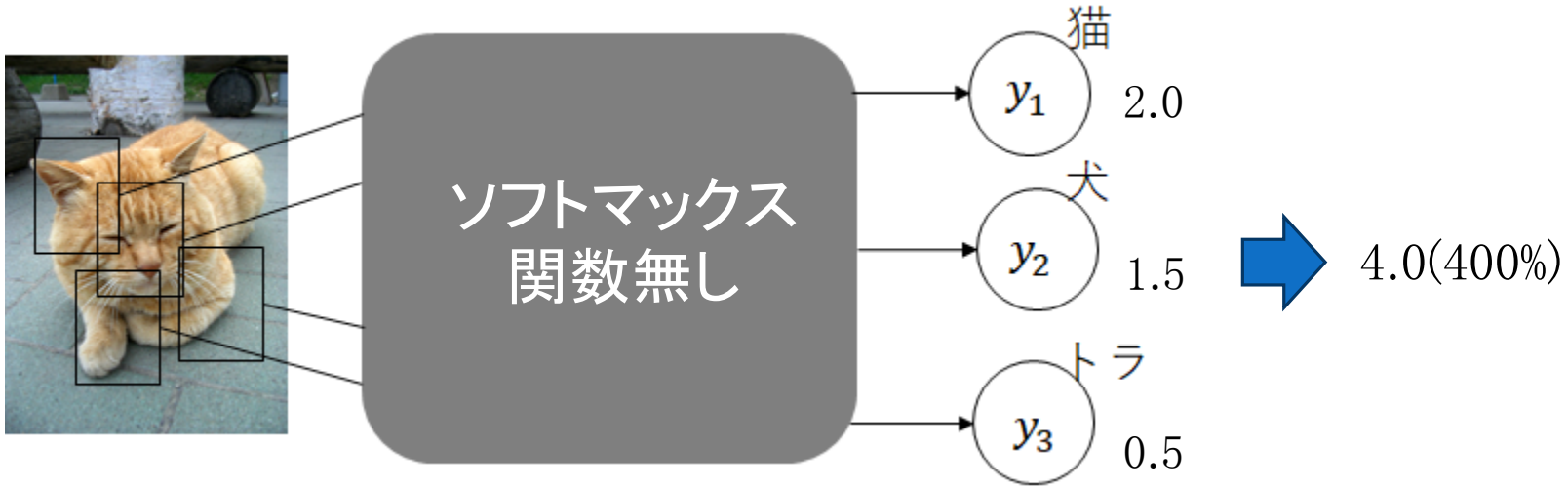


・出力を確率とみなせる
・分類問題に用いる

ソフトマックス関数の特徴

例:画像を見て猫と犬かトラを当てるプログラム

【ソフトマックス関数無し】

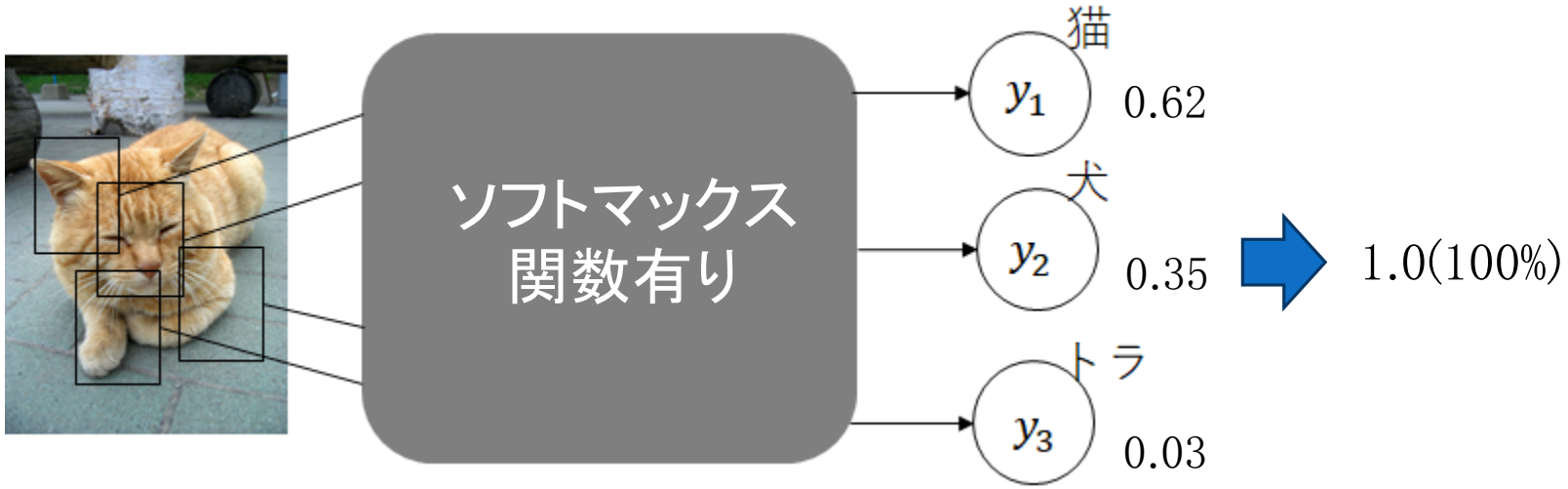


これは、全体の400%中、猫である確率が200%、犬である確率が150%、トラが50%言っているのと同じこと→わかりづらい

ソフトマックス関数の特徴

例:画像を見て猫と犬かトラを当てるプログラム

【ソフトマックス関数有り】

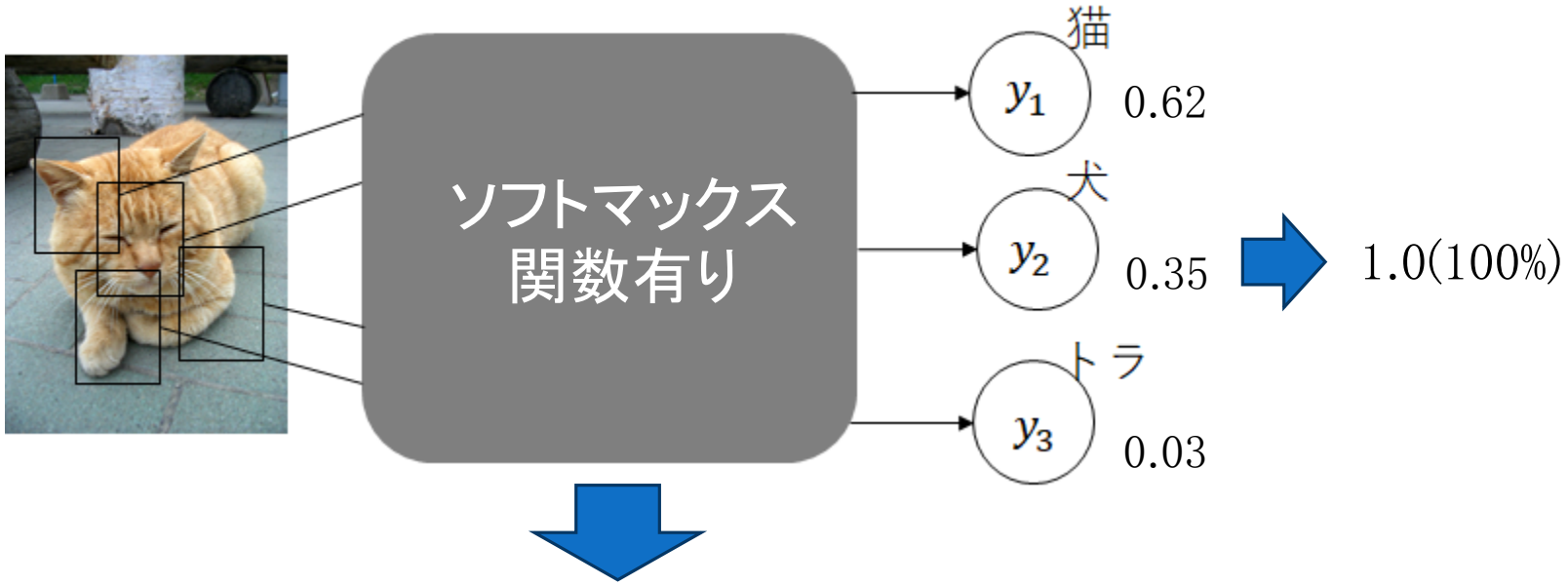


猫は0.62、犬は0.35、トラは0.03で、和が1.0になるようにソフトマックス関数は調整してくれます。
これだと全体の100%中、犬である確率が62%、猫である確率が38%となり、分かりやすくなる。

ソフトマックス関数の特徴

例:画像を見て猫と犬かトラを当てるプログラム

【ソフトマックス関数有り】



ソフトマックス関数は、出力されたごちゃごちゃした数字を綺麗に、全体の確率が100%(1.0)になるように調整してくれる関数。

ここまでが、前回お話しした内容です。

ここからは、ソフトマックス関数についてさらに知識が深まりそうなスライドをネットから抜粋しました。

最尤推定

- 最尤推定（さいゆうすいてい、英: maximum likelihood estimation、略してMLEともいう）
- ✓ 統計学において、与えられたデータからそれが従う確率分布の母数を点推定する方法である。（Wikipediaより）

- ✓ 僕はまだ、よくわからない....

最尤推定

最尤推定量とは？考え方を理解する
最尤推定量の定義に入る前に、まずは具体例をまじえつつその考え方を理解していきましょう。

最尤推定量とは、文字の如く、最も尤もらしい推定量のことです。このことから、なんとなく一番良い推定量だという気がしてきますよね？果たして本当にそうなのでしょうか。

ここで、「最も」は「一番」という意味ですが、では「尤もらしい」というのはどういう意味なのでしょうか？

ひとつ例を出して考えて見ましょう。

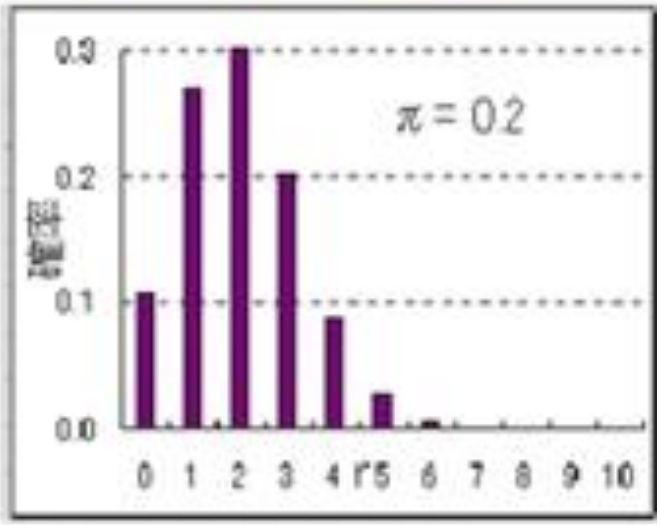
最尤推定

- 例えば、くじ引きでは、つぼの中の当たりくじの数は普通決まっている。
- ✓ 当たる回数と確率に一定のとびとびの関係があります。

- 右の棒グラフは、つぼの中に当たりくじが2割はいっている場合、10回引いてx回当たる確率を示したものです。

2回当たる場合が最も多いのが想像に難くないでしょう。

つぼの中に当たりくじが2割入っている場合



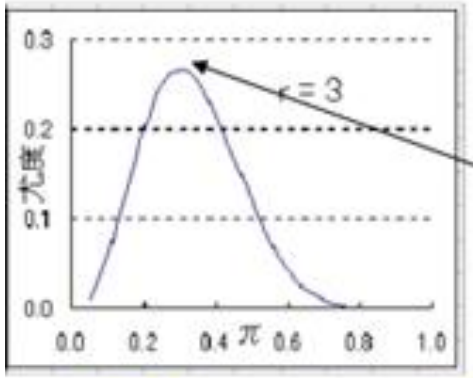
10回引いて2回当たる確率が最大

最尤推定

- たとえば当たる回数を3回に固定し、つぼの中の当たりくじの数を変えてみます（10個中1個～10個と変化させる）。
- つぼの中の当たりくじの数と確率が一定の連続関係をもちます（右の図）。このときの確率は、尤度という言い方をします。
- 尤度最大時の当たりくじ数を求めるのが最尤推定法です。

この場合、10回引いて3回当たる確率すなわち尤度は10個中3個入っている場合が最大となり、その確率は0.267となります。最尤推定量は0.3となります。

ここでいう尤度：
10回引いて3回当たる確率のこと



最大尤度は、

$$ {}_{10}C_3(0.3)^3(1-0.3)^7 $$
$$ \cong 0.267 $$

尤度は、π＝0.3、つまり
3割当たりくじの入っている
つぼを用いたとき最大になる。
最尤推定量＝0.3
これを求めるのが最尤推定法

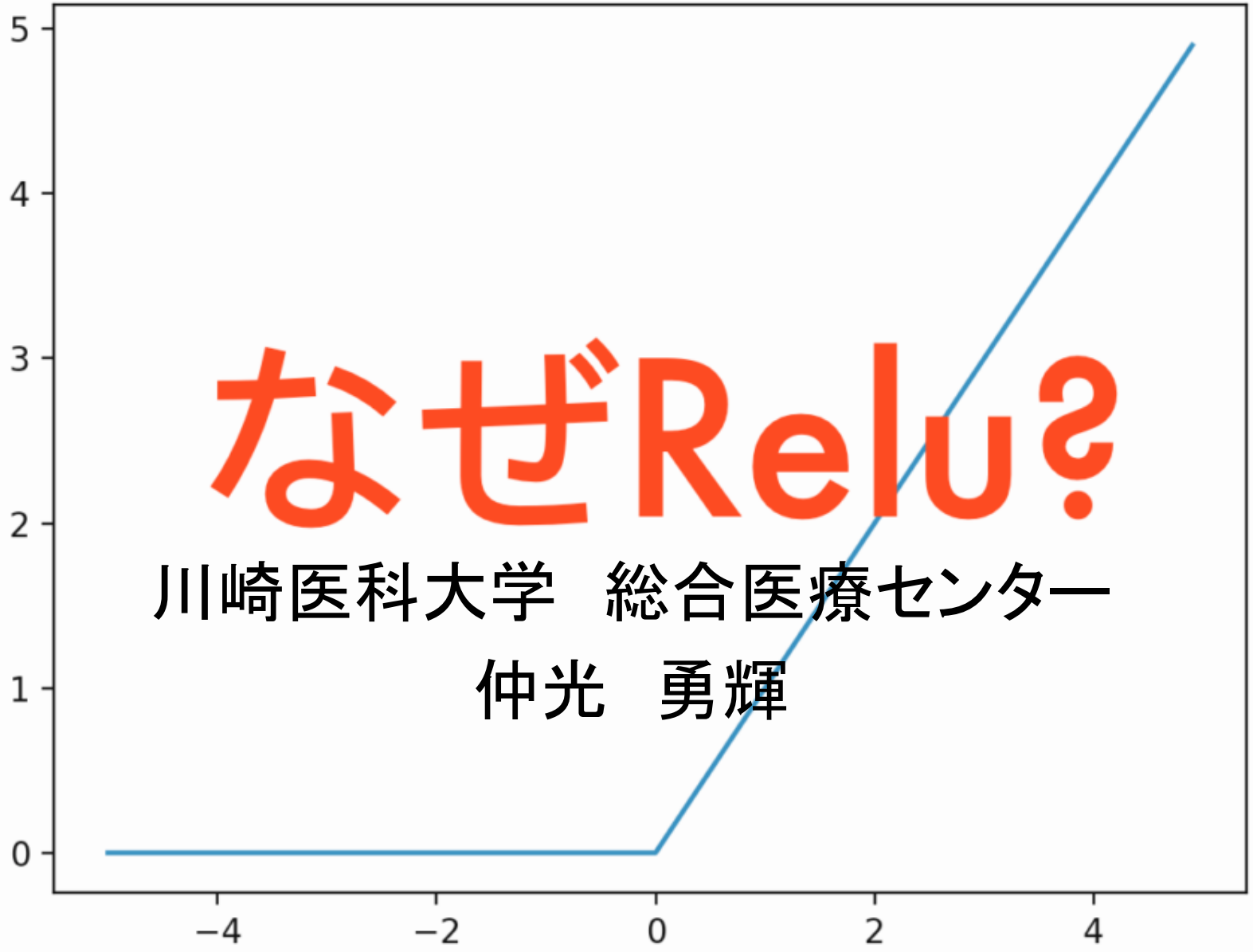
ここまでで僕が理解していること…

- 分類問題にソフトマックス関数を使用。
- ソフトマックス関数は出力の総和を1にしてくれる
  ありがたい関数。
- 出力を確率とみなせるようにする関数。
- ソフトマックス関数は最尤推定に関係がありそう。

以上4つです。

ご清聴ありがとうございました。

続きがあります



川崎医科大学　総合医療センター
仲光　勇輝

なぜReLU関数がいいの？？

まず前回のスライドから....

ReLU関数

【特徴】
・ReLU関数は最も一般的に用いられる活性化関数。

・入力が０を超えていれば、その入力をそのまま出力し、０以下なら０を出力する。

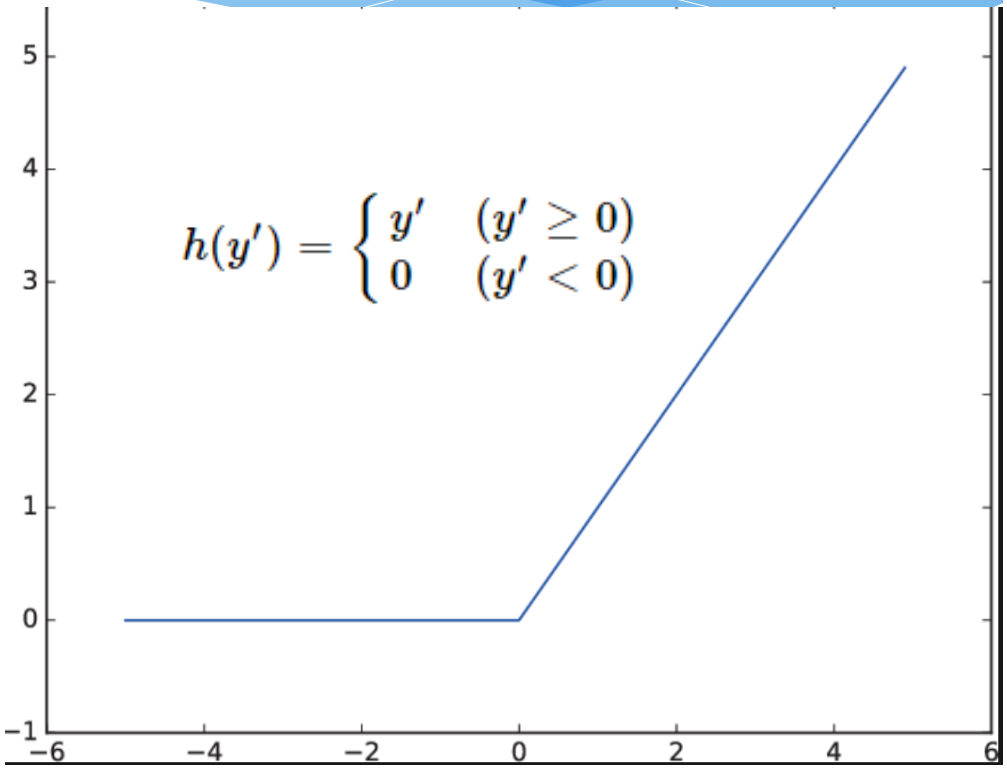
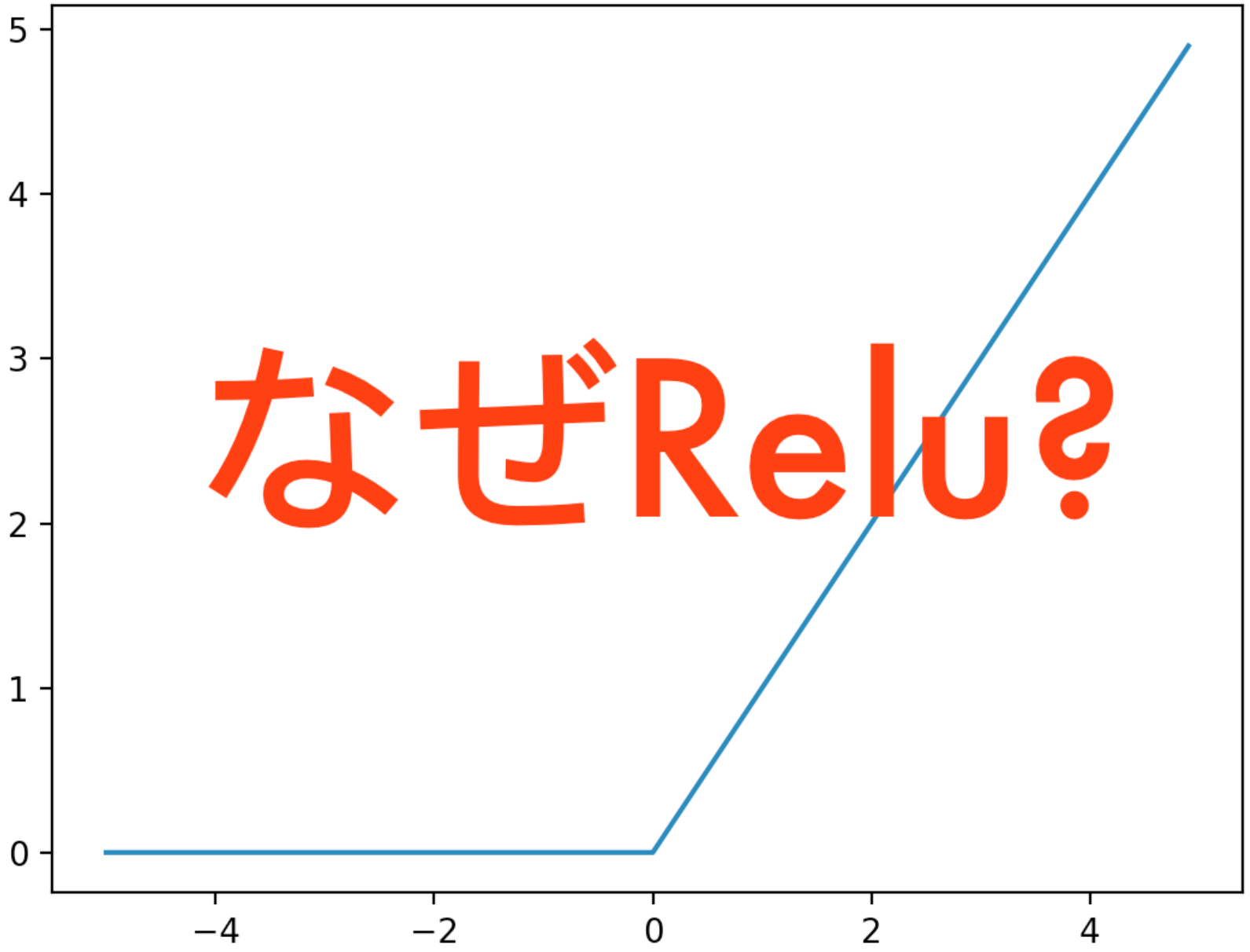


Fig11：ReLU関数


なぜRelu?
5
4
3
2
1
0
−4
−2
0
2
4

ReLU関数

- ReluはRectified Linear Unitの略称で,
正規化線形関数と日本語訳される。

ReLU関数

- この関数を言葉で説明すると…
入力値が0以下のとき0に,1より大きいとき入力値をそのまま出力する関数です。

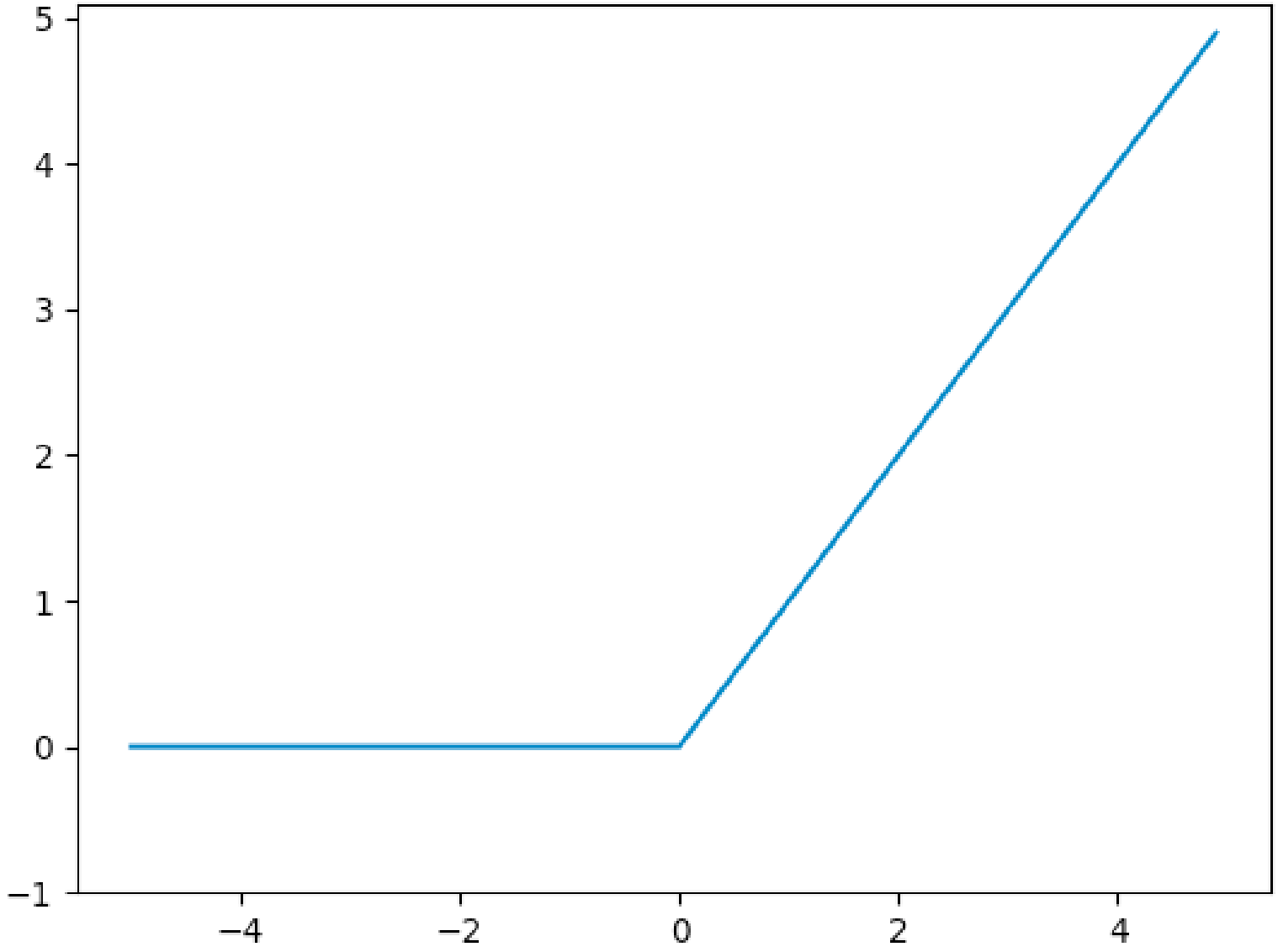
数式で表すと

$$f(x) = x_{x} = \max(0, x)$$

他サイトでみると…

$$R(x)=\max(x, 0)$$ と書かれていたり

図で表すと

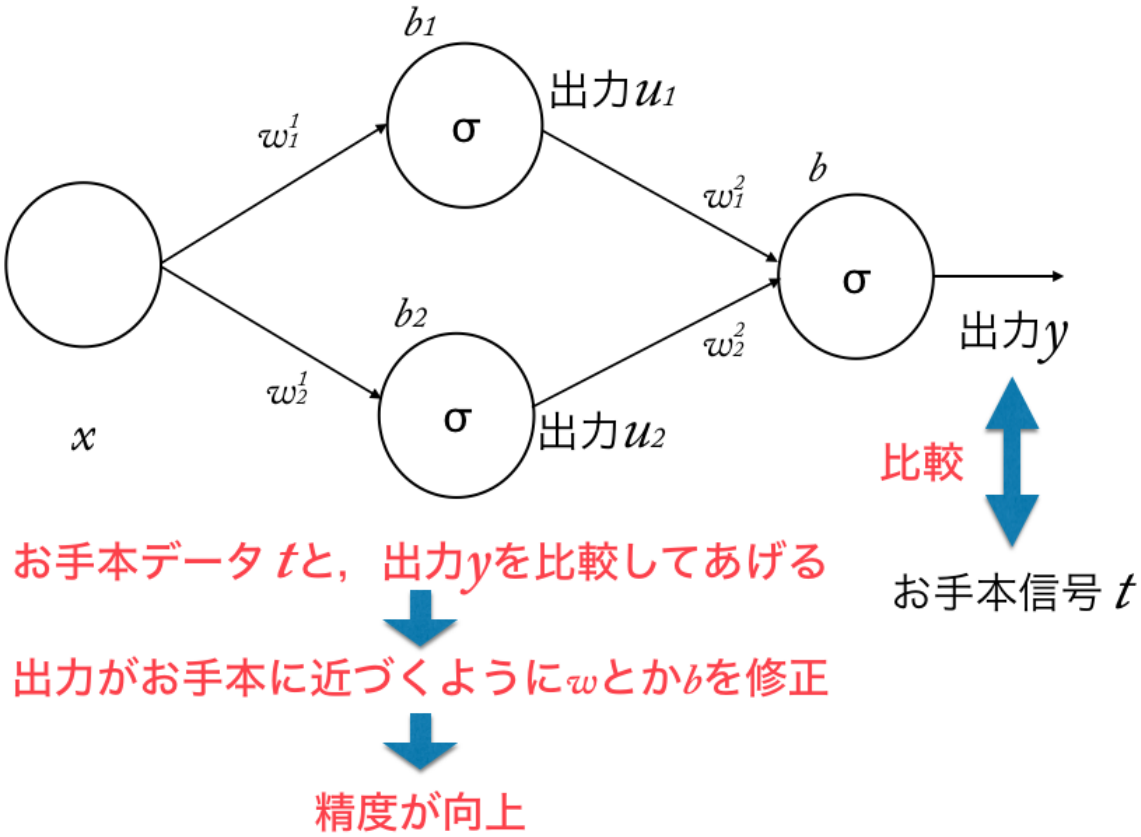


なぜReLUがいいのか??

- なぜReluがいいのか？
- では, なぜReluがディープラーニングで使われるのか？

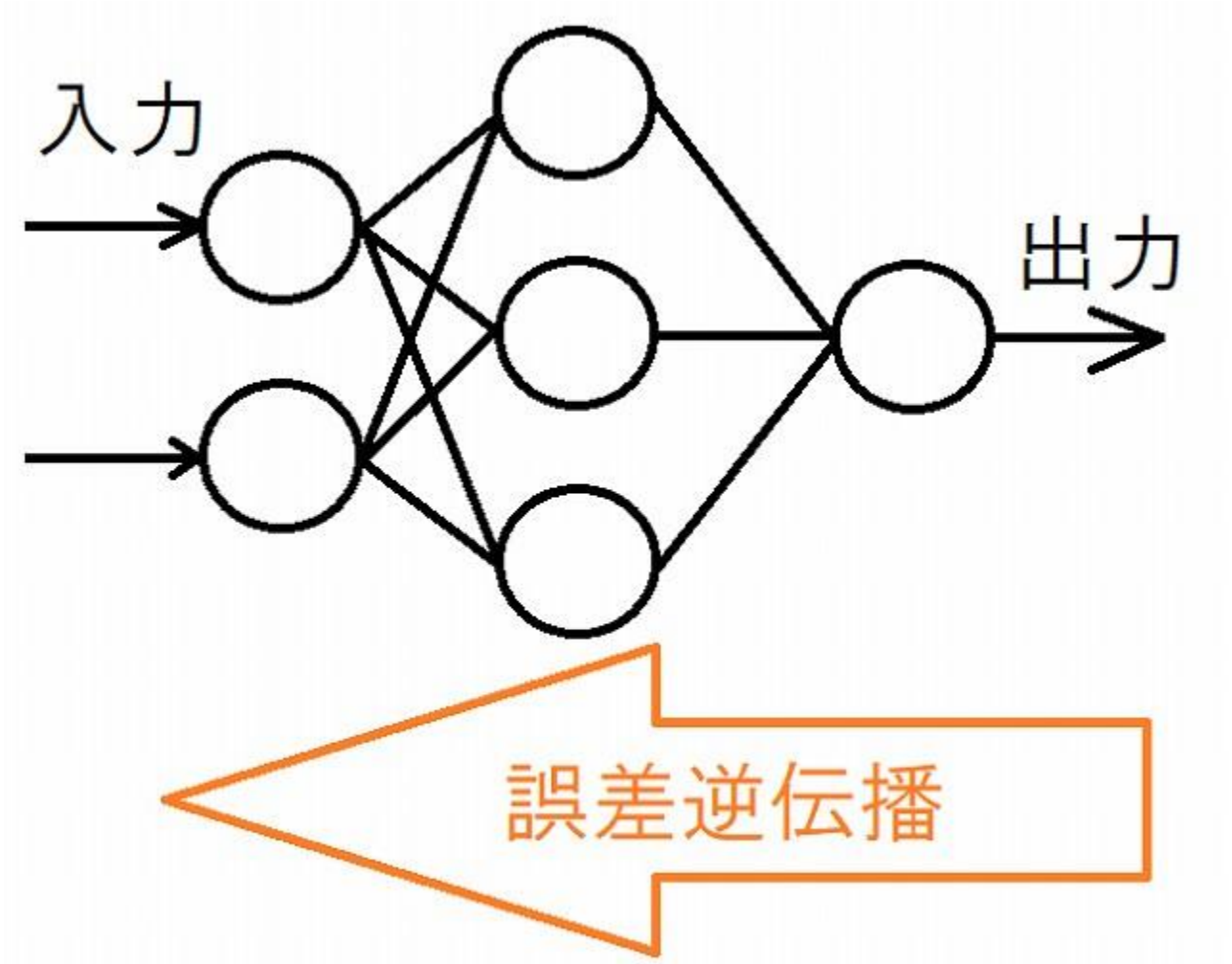
ディープラーニングではモデルのパラメータを学習するために、誤差逆伝搬法という方法を繰り返し用いています。

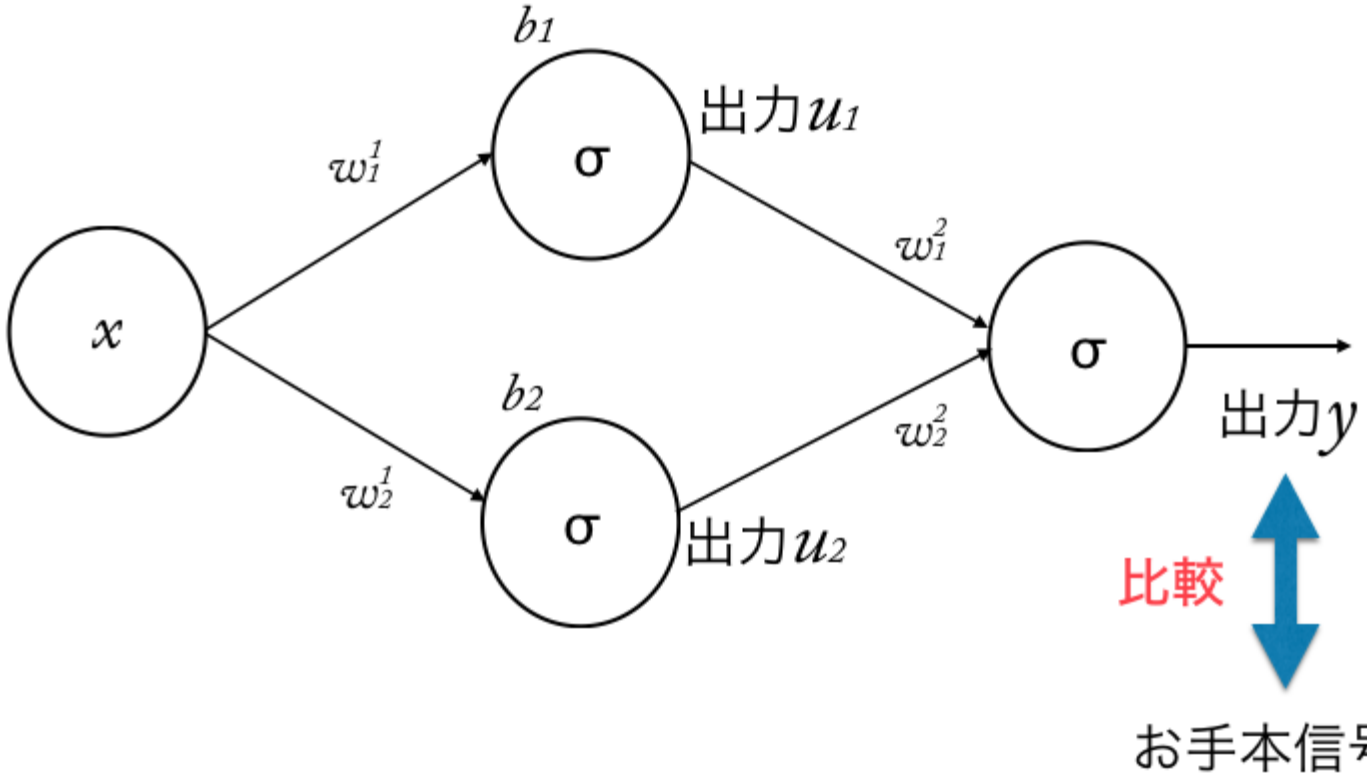
誤差逆伝搬法



- 御手本と出力を比較することで、重みwや、バイアスbを修正していく学習手法。

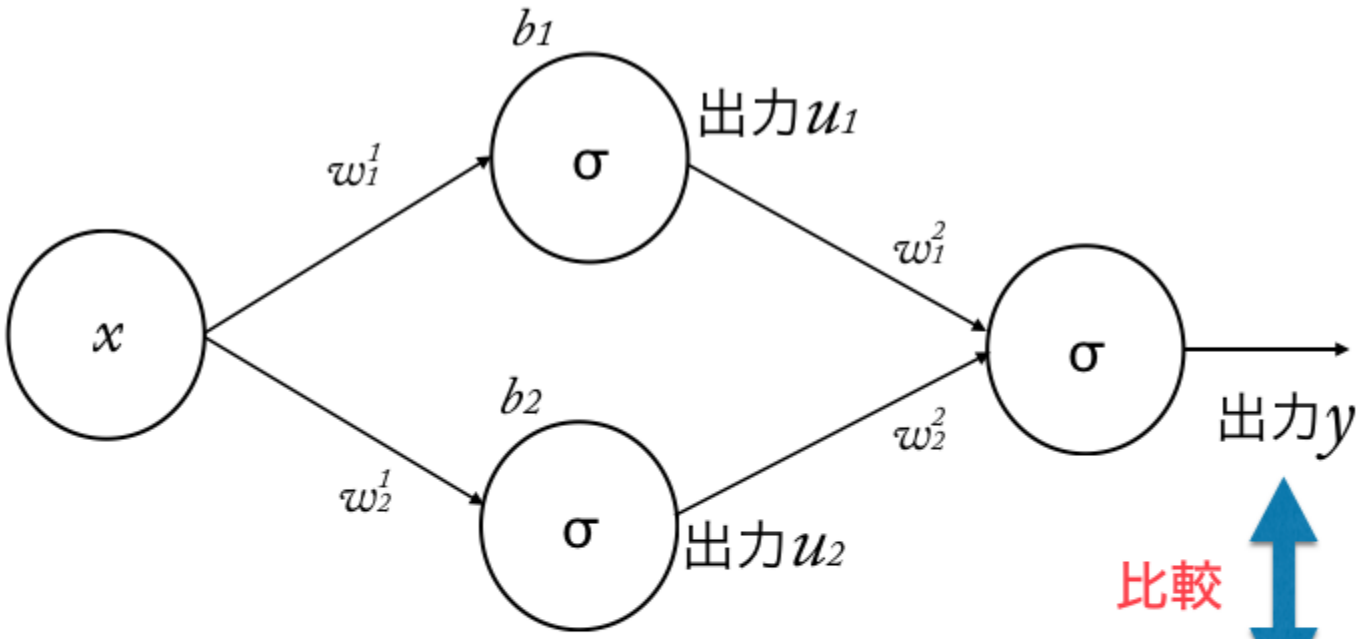
誤差逆伝搬法





お手本データ tと，出力yを比較してあげる

誤差Eは，出力yからお手本yを引いてあげると良い？



比較

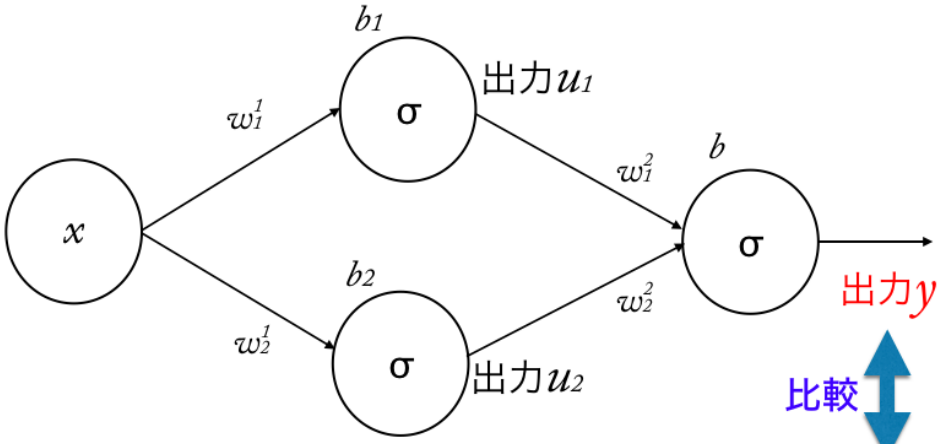
お手本信号 t

お手本データ tと，出力yを比較してあげる

誤差Eは，出力yからお手本yを引いてあげると良い？

正負考えるのめんどいから2乗を考える

- 誤差Eを，$E=(y-t)^2$と定義したい
- 今の目的は，重みwや，バイアスbを変化させることで出力信号とお手本信号の誤差をなくすこと
- つまり，重みwや，バイアスwの変化による，誤差Eの変化量が大事であり
- 変化量といえば… 微分.
- ということで，Eを微分することが多いので，$E=1/2(y-t)^2$と定義しておくほうが計算が楽である。
- あとはこの$E=1/2(y-t)^2$を0にするように重みwや、バイアスbを修正していくだけです.



お手本データ tと，出力yを比較してあげる

$$E=\frac{1}{2}(y-t)^2$$

エラー Eを0にするようにwやbを調整

ここまでが誤差逆伝搬法のお話

なぜReLUがいいのか

- なぜReluがいいのか？
- では, なぜReluがディープラーニングで使われるのか？

ディープラーニングではモデルのパラメータを学習するために、誤差逆伝搬法という方法を繰り返し用いています。

この誤差の微分が重要な点なのです！！

と書かれてありました

- Reluが導入される前は...
  - ✓ シグモイド関数、ステップ関数...

【欠点】

- 特定の微分値はとても小さくなる...
- その結果、誤差の微分がゼロになってしまう。
- そこで学習がストップしてしまう。
- そのため，学習モデルの性能があまり良くない。

- Reluはその関数の特性から, その微分の問題を解決しています。

- もう一度数式を見直すと、

- R(x)=max(x, 0)

→この微分は1となります
つまり、入力値が整数であればReluは、誤差の微分をゼロにすることなく、次の学習へつなげることができる。

活性化関数には他にもいろいろ種類があります.

しかしながら, reluがシンプルかつ協力であることから,

ディープラーニングではReluがよく用いられます。

まだまだ分からないことがたくさんありますが
Deep Learningの学習を日々続けていきます。

ご清聴ありがとうございました。