1. **入力層～中間層**
2. **ニューラルネットワーク**

**ニューラルネットワークは、「入力を線形変換する処理単位」がネットワーク状に結合した数理モデルである（wikiペディアより）。また、人間の脳をモデルとして作られた人工ニューロンをつなげたもので、ニューラルネットワークの構造には、入力層、中間層、出力層がある。**

**人工ニューロンとは、人間の脳神経系を模したニューラルネットワークを構成する基本単位。樹状突起に相当する複数の入力系統をもち、入力信号に重みを乗じた値の総和がある一定の閾値(いきち)を超えると、他のニューロンに信号を出力する。人工神経のことである（コトバンクより）。**

1. **事前に用意する情報**

**入力：**

**訓練データ：**

1. **多層ネットワークのパラメータ**

**活性化関数：**

**中間層出力：**

**総入力：**

**出力：=**

**誤差関数：**

**入力層ノードのインデックス：**

**中間層ノードのインデックス：**

**出力層ノードのインデックス：**

**層ノードのインデックス：**

**訓練データのインデックス：**

**試行回数のインデックス：**

* 確認テスト1
* ディープラーニングは、結局何をやろうとしているか2行以内で述べよ。

多数の中間層をもつニューラルネットワークを用いて、入力値から目的とする出力値に変換する数学モデルを構築すること。

* また、次の中のどの値の最適化が最終目的か。全て選べ。

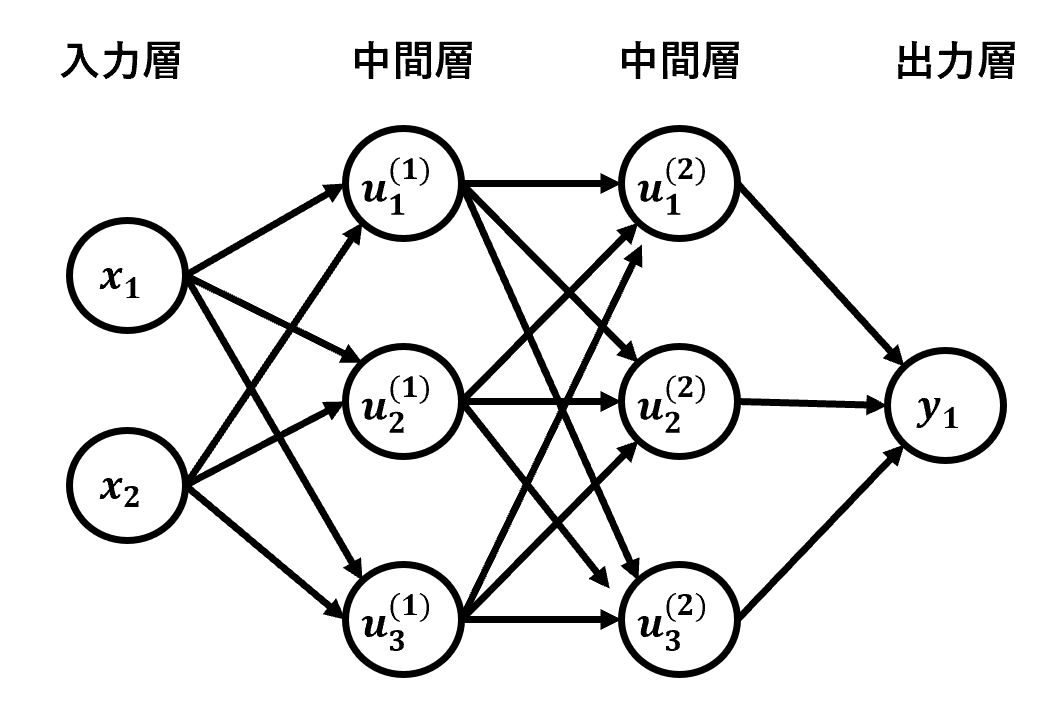
回答：③重み[W]、④バイアス[b]

* 確認テスト2
* 次のネットワークを紙にかけ。

入力層：2ノード1層

中間層：3ノード2層

出力層：1ノード1層



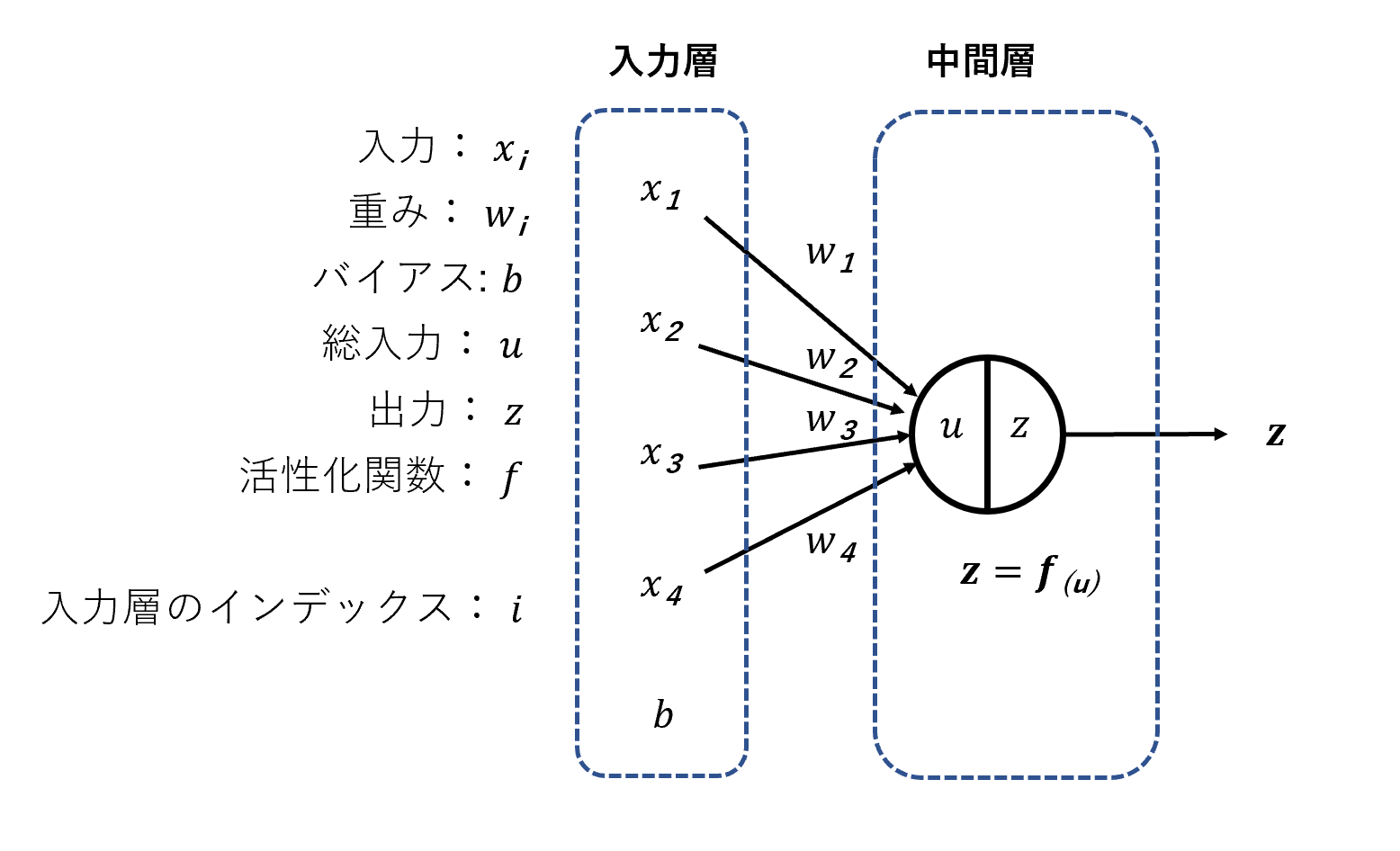
1. **入力層**

**人工ニューロンが最初に情報を受け取るのが入力層である。生き物のニューロンは電気信号によって情報を受け取るのに対し、人工ニューロンは数値で受け取る。**

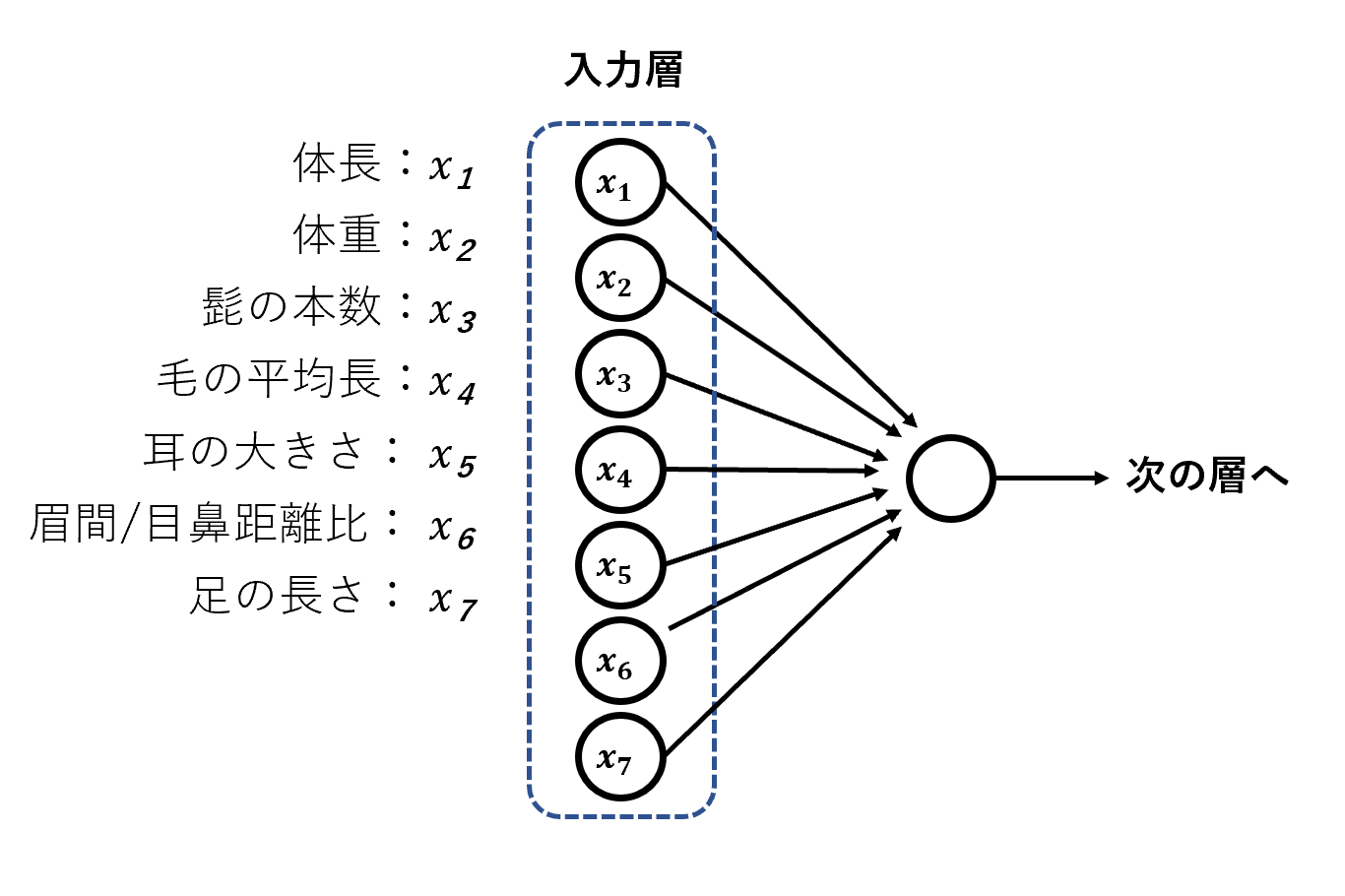
**​たとえば、手書きで「0」と書かれた画像を人工ニューロンに認識させる場合、入力層では画像の 1ピクセルを入力値として受け取る。入力層で受け取った情報は、ニューロン同士の結合の強度に応じて、優先順位が決定される仕組みである（sony ディープラーニングにおける中間層の役割とはより）。**

1. **中間層**

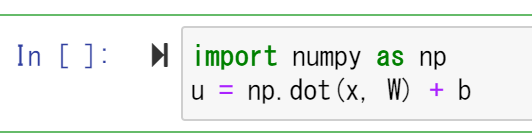
**入力層から情報を受け継ぎ、さまざまな計算を行うのが中間層である。中間層が多いほど複雑な分析ができ、中間層が3層以上あるニューラルネットワークをディープラーニングと呼ぶ。中間層の数に決まりはなく、扱う情報にあわせた任意での設定が可能である。なお、中間層は、隠れ層と呼ばれる場合もある（sony ディープラーニングにおける中間層の役割とはより）。**

****

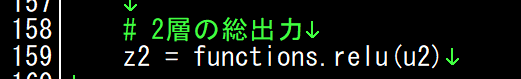
* 確認テスト3
* **この図式に動物分類の実例を入れてみよう。**

******

* 確認テスト4
* **この数式（1.2）をPythonで書け。**

******

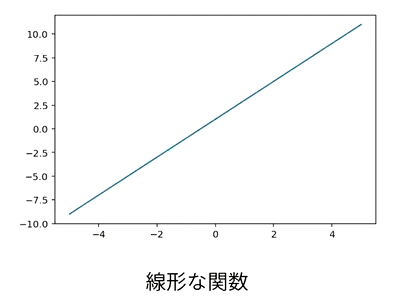
* 確認テスト5
* **「1\_1\_forward\_propagation.py」のファイルから中間層の出力を定義しているソースを抜き出せ。**

******

1. **活性化関数**

**ニューラルネットワークにおいて、次の層への出力の大きさを決める非線形の関数を表し、入力値の値によって、次の層への信号のON/OFFや強弱を定める働きを持つ。AI／機械学習のニューラルネットワークにおける活性化関数（Activation function）とは、あるニューロンから次のニューロンへと出力する際に、あらゆる入力値を別の数値に変換して出力する関数である。使われる活性化関数は時代とともに変化していて、ニューラルネットワークの基礎となっている情報処理モデル「パーセプトロン」では「ステップ関数」という活性化関数が用いられていた。しかし、「バックプロパゲーション」が登場してからは「シグモイド関数」や「tanh関数」が使われるようになった。さらに、最近のディープニューラルネットワークでは「ReLU」がよく使われるようになっている（@ITより）。**

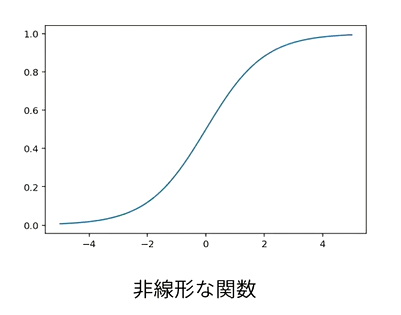
* 確認テスト6
* 線形と非線形の違いを図にかいて簡潔に説明せよ。



線形な関数は下記条件を満たす。

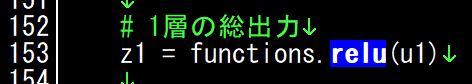
・**加法性：**

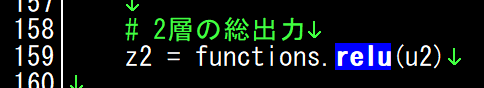
**・斉次性：**



**非線形な関数は加法性・斉次性を満たさない**

* 確認テスト7
* **配布されたソースコードより該当する個所を抜き出せ。**

****

****

1. **出力層**

**入力層と中間層で重みをかけ、活性化関数で処理された値が示されるのが出力層です。たとえば、手書きで書かれた「0」の画像について、何が書かれているのか判断した結果が出力されます。なお、出力層で得られた結果を教師データと照合し、出力層から入力層に向けて誤差の修正や調整を行う方法を「誤差逆伝播法」とよびます。これにより、多くの中間層をもつ複雑なニューラルネットワークでも、より適切な学習を行うことが可能です。出力層においてよく使われる主な活性化関数としては、下記があげられる。**

1. **分類問題（二値）の場合は「シグモイド関数」**
2. **分類問題（多クラス）の場合は「ソフトマックス（Softmax）関数」**
3. **回帰問題の場合は「（活性化関数なし）」もしくは（前述した）「恒等関数」**

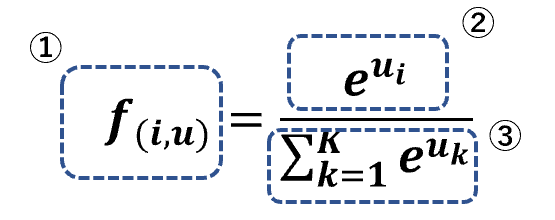
* 確認テスト８
* **なぜ、引き算でなく二乗するか述べよ。**

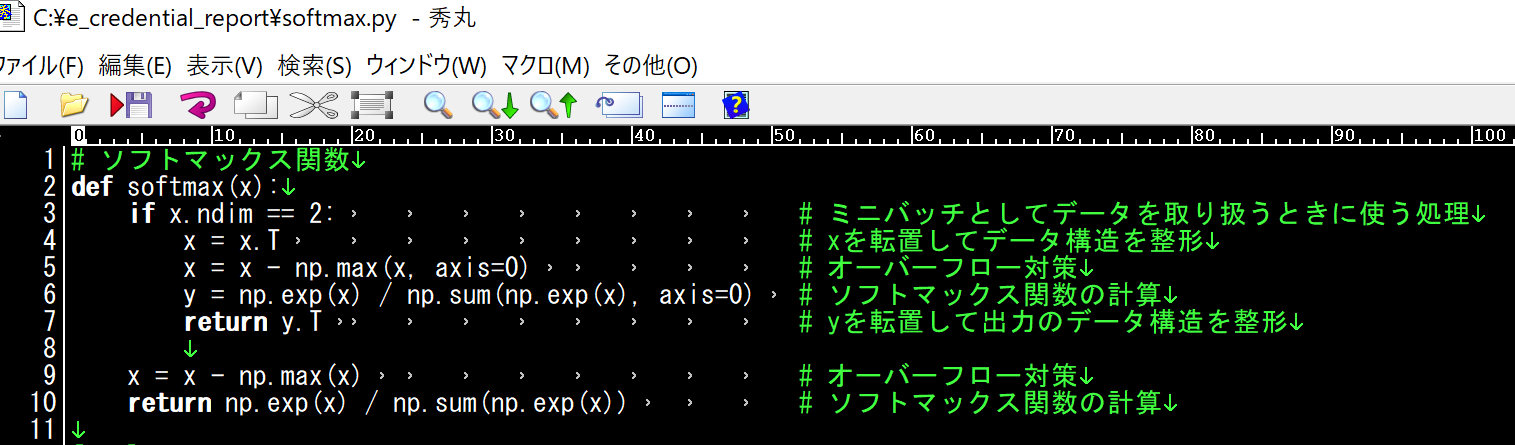
**引き算では、データの誤差の値で正負両方となり、全体の誤差を正しくあらわすのが難しい。２乗して各データの誤差を正の値になるようにし、計算を簡単にするため。**

* **下式の1/2はどういう意味を持つか述べよ。**

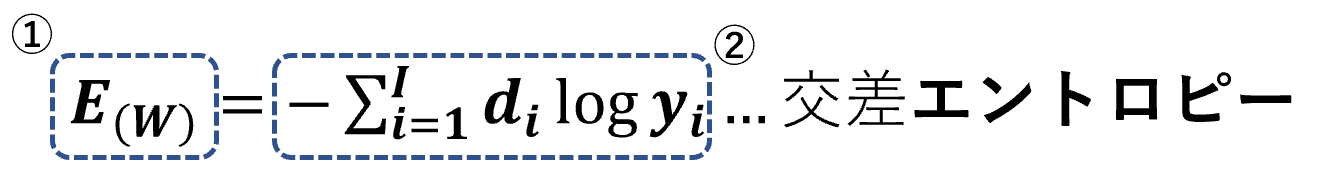
**目的関数を微分する際に計算式を簡単にするため。**

* 確認テスト9
* **①～③の数式(下記)に該当するソースコードを示し、一行づつ処理の説明をせよ。**

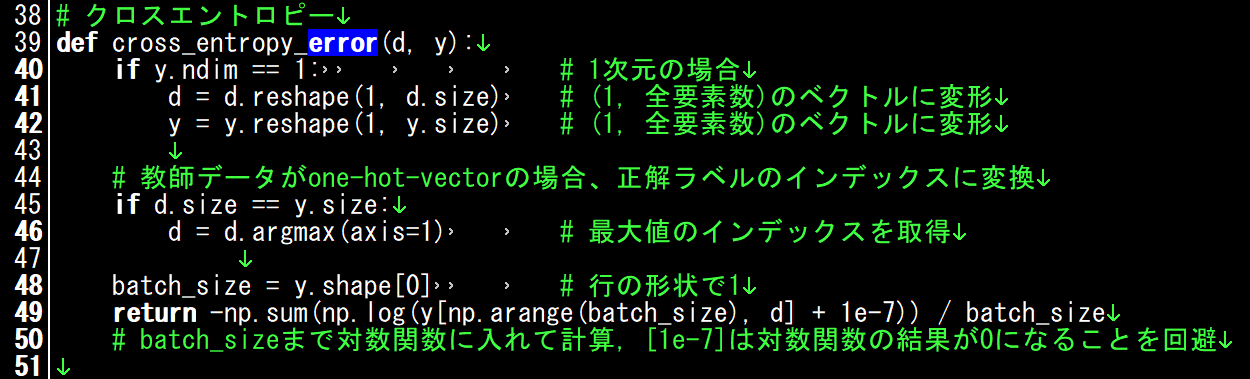
****

****

* **確認テスト10**

****

* **①、②の数式に該当するソースコードを示し、一行づつ処理の説明をせよ。**

****

1. **勾配降下法**

**訓練テストに対してコストが最小になるように、モデルパラメータを少しづつ操作し、モデルを訓練テストに対して適合したパラメータに収束させる方法で、下記の降下法が代表的である（Qiitaより）。**

1. **勾配降下法**

**勾配降下法はコスト関数を最小にするためにパラメータを繰り返し操作して最適なパラメータを算出する。**

**パラメータベクトルについて誤差関数の局所的な勾配を測定し、降下の方向に進めていき、勾配が0になればコスト関数を最小にするパラメータを見つけることができたことになる。**

1. **確率的勾配降下法（SGD）**

**確率的勾配降下法（SGD）とは、連続最適化問題に対する勾配法の乱択アルゴリズムのことで目的関数が、微分可能な和の形である事が必要である。バッチ学習の最急降下法をオンライン学習に改良した物である。**

1. **ミニバッチ勾配降下法**

**ミニバッチ勾配降下法は一度に利用する訓練データの量がバッチ勾配降下法と比較して少なくなっている（ミニ）ことから命名されており、訓練データの中からいくつかのデータを取り出し、そのデータで計算した勾配に基づいてパラメータを更新する手法である。バッチ勾配降下法と比較して計算に必要なメモリ量が少ないこと、確率的勾配降下方と比較して外れ値の影響を受けにくく学習が比較的安定して進むという利点がある（AIZINEより）。**

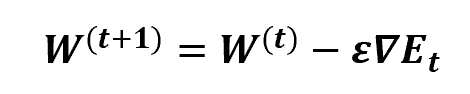
* **確認テスト11**
* **該当するソースコードを探してみよう。**

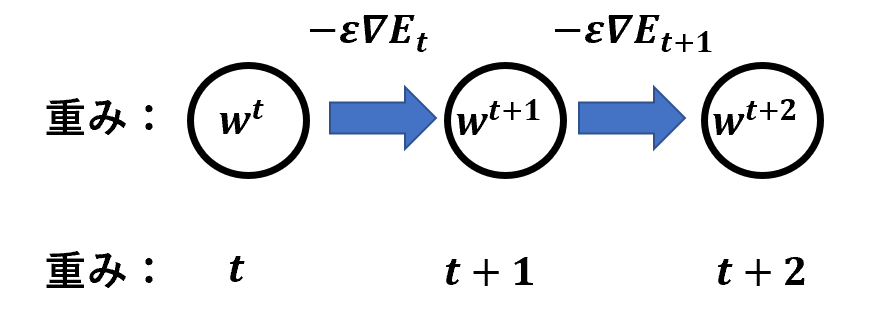
****

* **確認テスト12**
* **オンライン学習とは何か2行でまとめよ。**

**学習データが入ってくる毎にその都度パラメータを更新して学習を進めていく方法。**

* **確認テスト13**
* **この数式の意味を図に書いて説明せよ。**

****

****

1. **誤差逆伝播法（バックプロパゲーション）**

**誤差逆伝播法は、損失関数をゼロに近づけるために必要なパラメータの勾配（偏微分値）を、効率的に求める計算方法である（Qiitaより）。**

**ニューラルネットの出力を個のパラメータの下記関数として表す。**

**次に損失関数（Loss関数）をとし、を用いてこの推論値と教師データとの誤差を測る。**

**Deep Learningの学習においては、 の値がゼロになる（近づく）ようにパラメータを決めたい。この時以下のような勾配（偏微分のベクトル）が、Lの極大値の方向を指し示すベクトルであることが、ベクトル解析の分野で知られているところです。ですから現在のパラメータの地点から勾配の逆方向にパラメータを更新すれば、Lは限りなくゼロに近づいて行くことが期待できます。つまりパラメータ（W)から勾配(xΔ)を引いて更新します。Δは学習率と呼ばれ、非常に小さい値(0.01とか)をとればよいことが知られています。このようにすればWは勾配と逆方向に少しだけズレることになります。**

**パラメータの更新は以下の式で表される。**

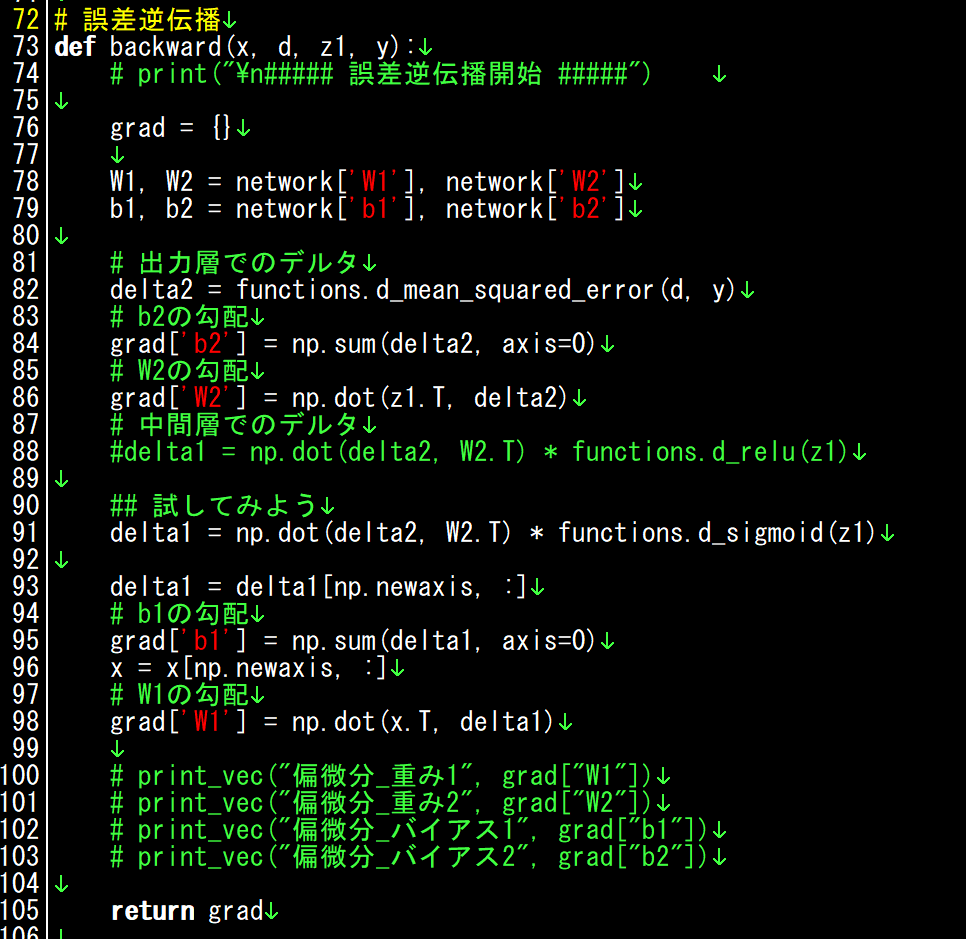
**勾配を求めるためには、数値微分を使い、任意のパラメータを一つ取り、小さな値だけずらして、Lの値を再計算し、傾きを求める。誤差逆伝播法は、ニューロンネットワークの逆方向に合成関数の偏微分の公式を当てはめていくことで、勾配の計算を効率的に行う方法である（Qiitaより）。**

1. **合成関数の偏微分の公式**

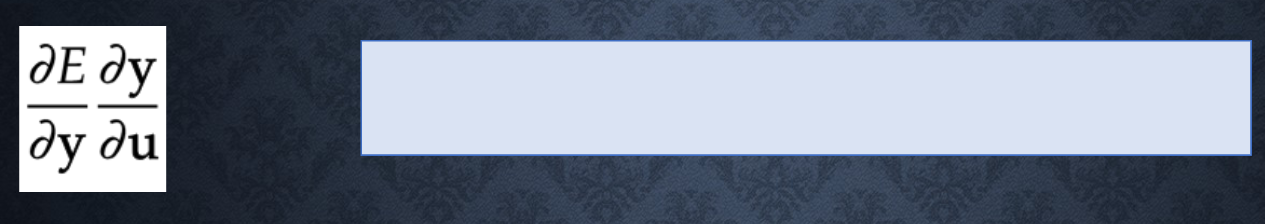
**を変数の関数とする。**

**を変数の関数とする。**

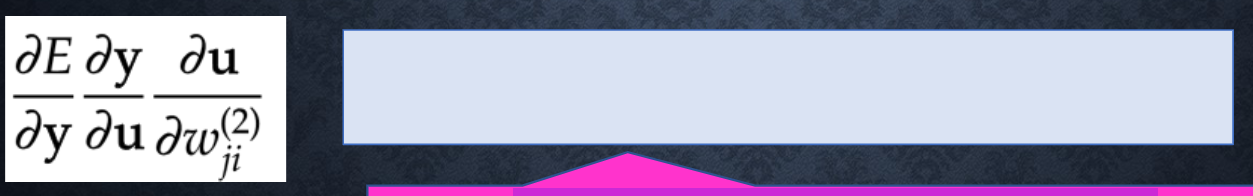
* **確認テスト14**
* **誤差逆伝播法では不要な再帰的処理を避ける事が出来る。既に行った計算結果を保持しているソースコードを抽出せよ。**

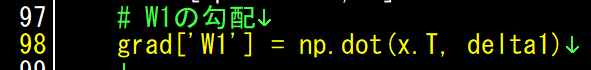
****

* **確認テスト15**
* **2つの空欄に該当するソースコードを探せ。**

****

****

****

****