# １　Section1\_入力層~中間層

ニューラルネットワークの構造（２層）は以下のようになる。

　　 入力層　　　中間層　　出力層

　　　　　　　　　　　　　X0　　　　Z0　　　　Y0

　　　　　　　　　　　　　X1 Z1　　　　Y1

　　　　　　　　　　　　　X2 Z2　　　　Y2

　入力層X0,X1,X2として、中間層へはW00・X0、W01・X1、W02・X2を渡す。

　入力総和Z0＝W00・X0＋W01・X1＋W02・X2

　入力総和Z1＝W10・X0＋W11・X1＋W12・X2

　入力総和Z2＝W20・X0＋W21・X1＋W22・X2

　行列式より

　　　W00　W01　W02　X0　　Z0

　　　W10　W11　W12　X1　= Z1

　　　W20　W21　W22　X2　　Z2

# ２　Section2\_活性化関数

ニューラルネットワークにおける活性化関数とは、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。

１）ReLU

　関数への入力値が０以下の場合は、出力値が常に０、入力値が０より大きい場合は、出力値が入力値と同じになる。

２）Mish

　関数への入力値が０以下の場合は、出力値がほぼ０、入力値が０より大きい場合は、出力値が入力値と同じになる。

３）活性化関数一覧

|  |  |  |
| --- | --- | --- |
| No | 活性化関数 | 概要 |
| 1 | ステップ関数 | Y= 1(x>0)  0(x≦0) |
| 2 | 恒等関数 | Y=x（順伝播）  δｙ/δｘ＝１（逆伝播） |
| 3 | Bent Idenntity | ｙ＝1/2(√x2+1-1)+x |
| 4 | HardShirink | Y=x(x＜-λ or λ＜x)  0 |
| 5 | SoftShrink | Y=x+λ(x＜-λ)（順伝播）  　x－λ(λ＞x)  　0  δｙ/δｘ＝１（逆伝播）  　　　　　0 |
| 6 | Threshold | Y=x（順伝播）  0  Y=1（逆伝播）  0 |
| 7 | シグモイド関数 | Y=1/(1+e-x) （順伝播）  δｙ/δｘ＝y(1-y)（逆伝播） |
| 8 | HardSigmoid | Y=1  0.2x+0.5  0  δｙ/δｘ=0.2  0 |
| 9 | logSigmoid | Y=log(1/(1+e-x)  δｙ/δｘ=1/(1+ex) |
| 10 | Tanh | Y=tanh(x)=(ex-e-x)/(ex+e-x)  δｙ/δｘ=sech2(x)=1/cosh2(x)=4/(ex+e-x)2 |
| 11 | tanhSirink | Y=x-tanh(x)  δｙ/δｘ=tanh2(x) |
| 12 | Hardtanh | Y=1  -1  X  δｙ/δｘ=0  1 |
| 13 | ReLU関数 | Y=x(x＞0)  0(x≦0)  δｙ/δｘ=1(x＞0)  0(x≦0) |
| 14 | ReLU6 | Y=0(x≦0)  6(x≧6)  X  δｙ/δｘ=0((x≦0) or (x≧6))  1 |
| 15 | Leak-ReLU | Y=x(x＞0)  0.01x(x≦0)  δｙ/δｘ=1(x＞0)  0.01(x≦0) |
| 16 | ELU | Y=x (x≧0)  α(ex-1) (x＜0)  δｙ/δｘ=1  αex |
| 17 | SELU | Y=λx  λα(ex-1)  δｙ/δｘ=λ  λαex |
| 18 | CELU | Y=x  α(e(x/α)-1)  δｙ/δｘ=1  e(x/α) |
| 19 | ソフトマックス関数 | Yi=exi/Σexk  δｙ/δｘ= yi(1-yi)Σyiyj=Σyi(δij-yj) |
| 20 | Softmin | Yi=e-xi/Σe-xk  δｙ/δｘ=yi(1-yi)+Σyiyj=Σyi(δij-yj) |
| 21 | Logsoftmax | Yi=log(exi/Σexk)  δｙ/δｘ=Σ(δij-eyj) |
| 22 | Softplus | Y=log(1+ex)=ln(1+ex)  δｙ/δｘ=ex/(1+ex)=1/(1+e-x) |
| 23 | Softsign | Y=x/(1+|x|)  δｙ/δｘ=1/(1+|x|)2 |
| 24 | Swish | Y=x/(1+e-βx)  δｙ/δｘ=βy+(1-βy)/(1+e-βx) |
| 25 | hardSwish | Y=0  X(x+3)/6  X  δｙ/δｘ=0  (2x+3)/6  1 |
| 26 | ACON |  |
| 27 | Mish |  |
| 28 | tanhExp | Y=xtanh(ex)  δｙ/δｘ=tanh(ex)-xex(tanh2(ex)-1) |

# ３　Section3\_出力層

　　 入力層　　　　　 出力層

　　　　　　　　　　　　　X1　　　　　　　　u11|

　　　　　　　　　　　　　X2

　　　　　　　　　　　　　X3

出力層では、活性化関数を使用して非線形変換を行う。

出力層の設計

　１）ニューラルネットワークは分類、回帰の両方に用いられる。

　２）どちらの問題を解決するかで、活性化関数を変更する必要がある。

　３）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用いる。

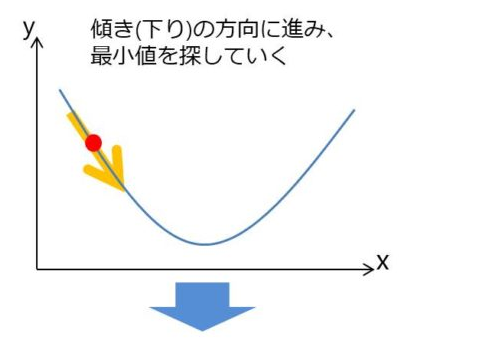
４）クラス分類では、出力層のニューロンの数はクラス数

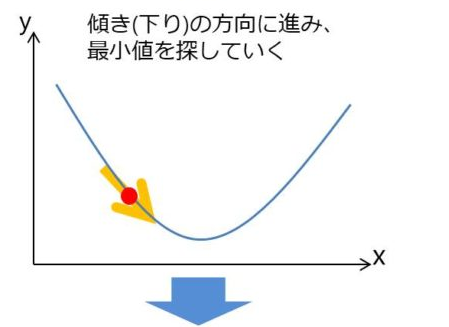
# ４　Section4\_勾配降下法

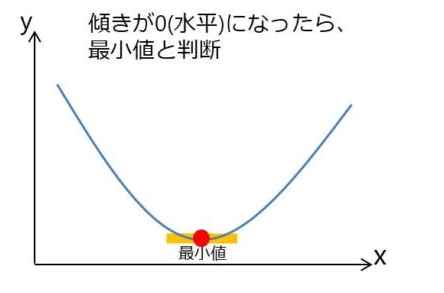
　モデルに対してコストが最小になるようにパラメータを少しづつ変化させて、トレーニングに適合したパラメータを算出するアルゴリズムである。

ｘ＝ｘ－η（ｄｆ（ｘ）／ｄｘ）

１）連鎖律



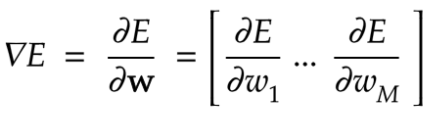




|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 手法 | 利用データ | 計算時間 | メリット | デメリット |
| バッチ勾配降下法 | 全てのデータ | 大 | ・解への到達が早い  ・結果が安定 | ・メモリの使用量が多い  ・局所解にはまりやすい |
| SGD | １つのデータ | 小 | ・メモリの使用量が少ない  ・オンライン学習が可能  ・局所解を回避する可能性がある | ・解への到達が遅いことがある  ・はづれ値の影響を大きく受ける |
| ﾐﾆﾊﾞｯﾁ勾配降下法 | 一部のデータ | 中 | ・バッチ勾配降下法とSGDのそれぞれのメリットがある | それぞれのデメリットがある |

# ５　Section5\_誤差逆伝播法

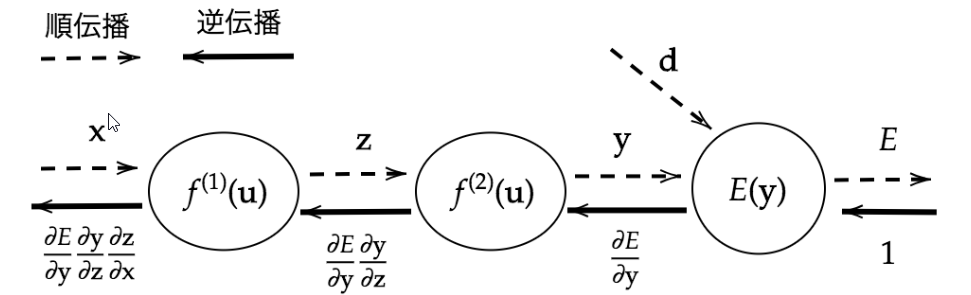
（1）誤差逆伝播法の概要

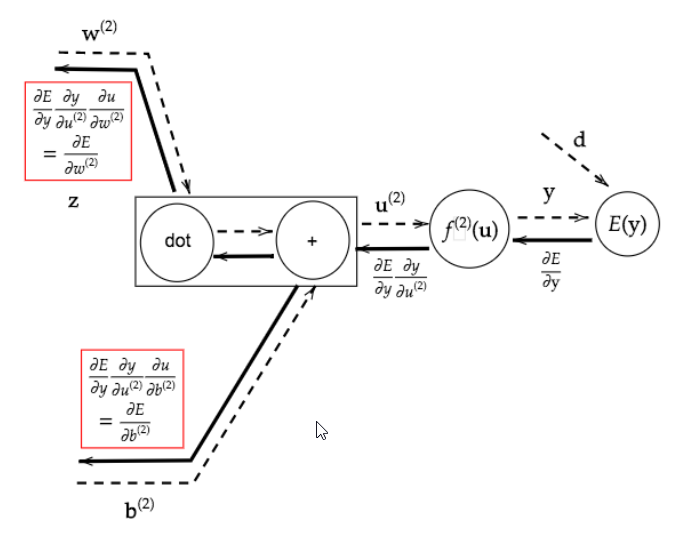


【誤差逆伝播法】

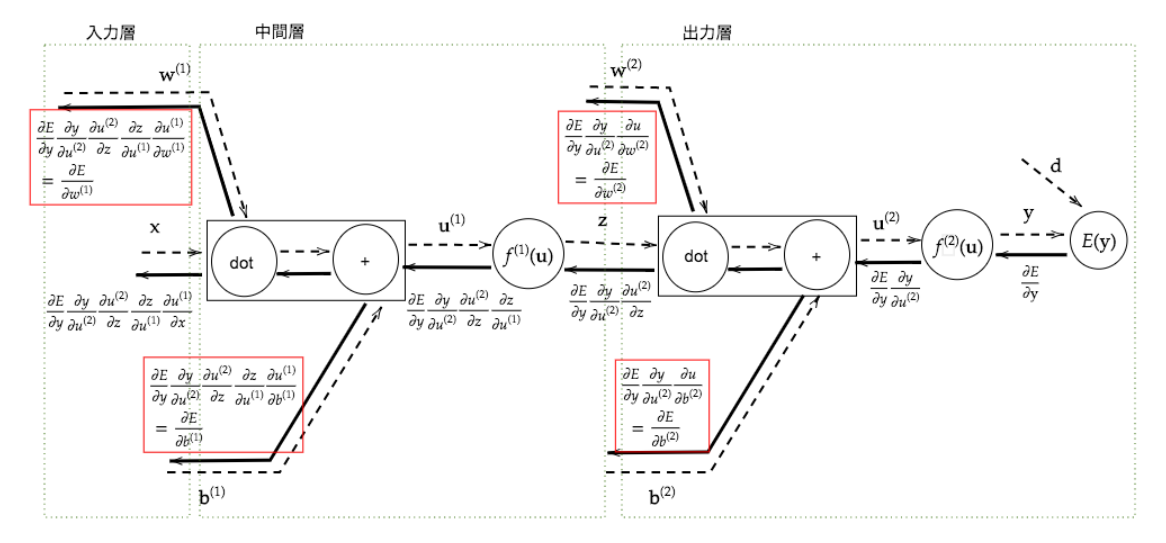
　算出された誤差を、出力層側から順に微分し、前の層前の層へと伝播させる。

　最小限の計算で各パラメータでの微分値を解析的に計算する手法である。





計算結果（=誤差）から微分を逆算することで、不要な再帰的計算を避けて微分を算出する。



計算結果（=誤差）から微分を逆算することで、不要な再帰的計算を避けて微分を算出できる。

（1）誤差勾配の計算について

