3章　確率的勾配法

内容
機械学習における、学習時のテクニックをまとめた章

3.1 確率的勾配法(SGD)

3.2 汎化性能と過剰適合

3.3 正則化

3.4 学習率の選定と制御

3.5 SGD

3.6 層出力の正規化

3.7 重みの初期化

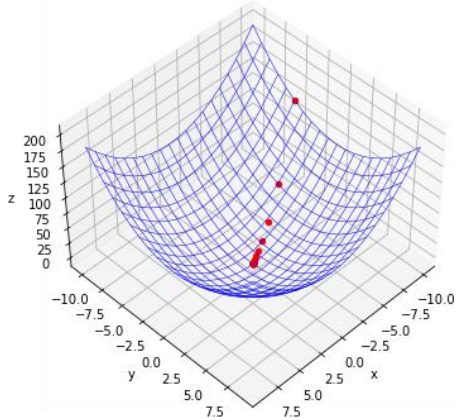
3.1.1 勾配降下法の基礎

機械学習の学習は、ニューラルネットワークの最適なパラメータを探索することが目的。
※最適なパラメータとは損失関数を最小にするパラメータの値（重み,バイアス）。

ここで損失関数は一般的に複雑なために最適なパラメータを直接求められません。
そのため損失関数の極小値を求めるために勾配降下法が利用される。
勾配降下法では、現在のパラメータで最も損失関数を減らす方向を示す**勾配方向**、一定の距離（学習率）ずつパラメータを変化させる。そして移動したパラメータでも繰り返し勾配方向に移動します。この繰り返しによって損失関数の最小値を求めます

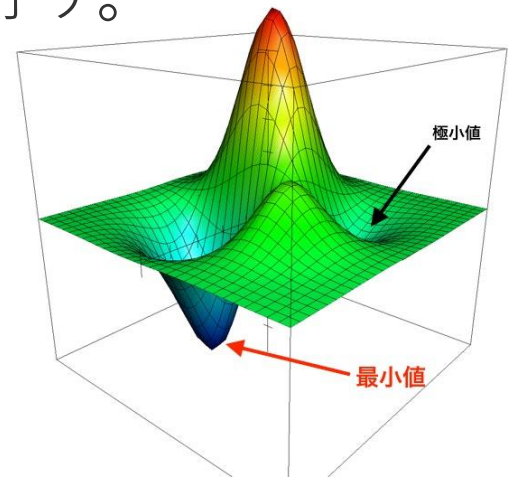
勾配　$$\nabla E = \frac{\delta E}{\delta \boldsymbol{w}} = [\frac{\delta E}{\delta w_1} \cdots \frac{\delta E}{\delta w_M}]^T$$

更新式　$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \epsilon \nabla E$$



3.1.2 バッチ学習

バッチ学習は全部のサンプルを利用して1回のパラメータ更新を行う。
バッチ:全サンプルのこと



損失　　$E(\boldsymbol{w}) = \sum_{n=1}^{N} E_n(\boldsymbol{w})$　　全てのサンプルを参照

更新式　$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \epsilon \nabla E$

欠点:最適解ではない極小値に陥ってしまった場合抜け出せない

3.1.2 バッチ学習からSGDに変える利点

バッチ学習の最適解ではない極小値に陥ってしまった場合抜け出せない欠点を解決するために提案されたのが確率的勾配法(SGD)である。

バッチ学習とSGDの違いは、パラメータの1回の更新に全サンプルではなく、ランダムにピックアップした1つのサンプルを使うということだけです。

毎回ランダムに違うデータを使っているため、1つ前のサンプルでローカルに陥ったとしても次にランダムに選んだサンプルでは損失が変化するため、再びパラメータが大きく更新され極小値から脱出できます。

更新式　$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \epsilon \nabla E_n$

欠点: 1つのサンプルによる勾配で更新しないと次のサンプルには移れないため、並列化ができません。

3.1.3 SGDにミニバッチを使う利点

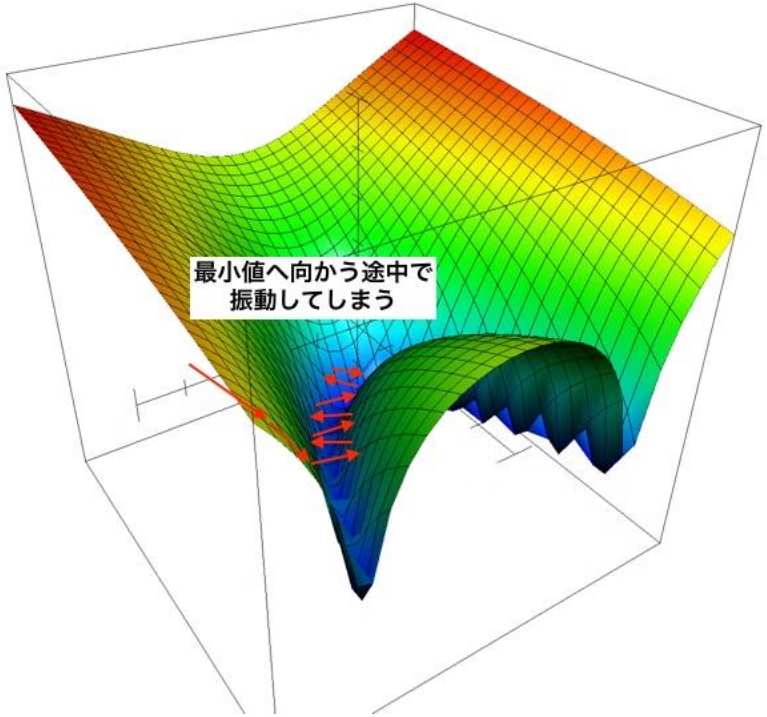
SGDの欠点である並列計算できない点を解決するために提案されたのがミニバッチの利用である。

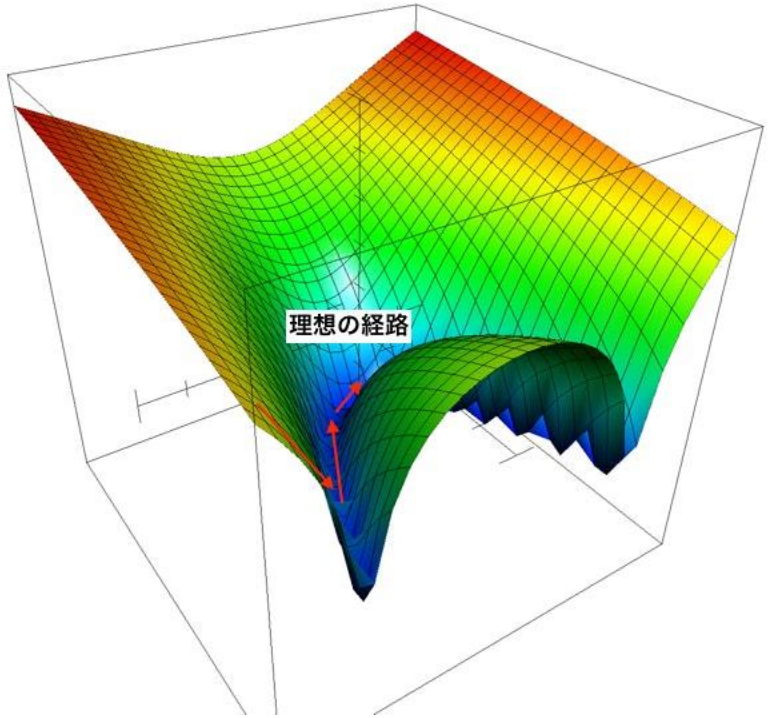
更新で使うサンプル数が1つだからこそ並列化ができないのだから、このサンプル数を増やせばバッチ学習のように並列化ができるという考えです。ただ全てのサンプルを利用するとバッチ学習となるので、サンプルの一部の集合（ミニバッチ）で学習を行います。

損失　　$E_t(\boldsymbol{w}) = 1/N_t \sum_{n \in D_t} E_n(\boldsymbol{w})$

3.1.4 モーメンタム

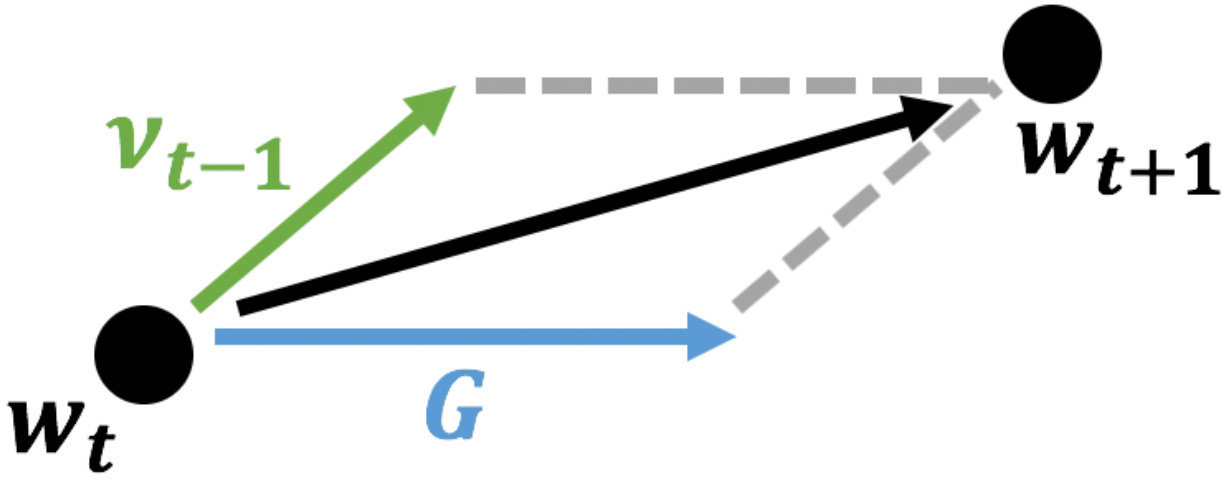
損失関数の最小値にたどり着きたいというモチベーションがある中で、
鋭いくぼみを持つ形状があり振動を起こしてしまうため最小値にたどり着くのが遅くなることがあります。





3.1.4 モーメンタム

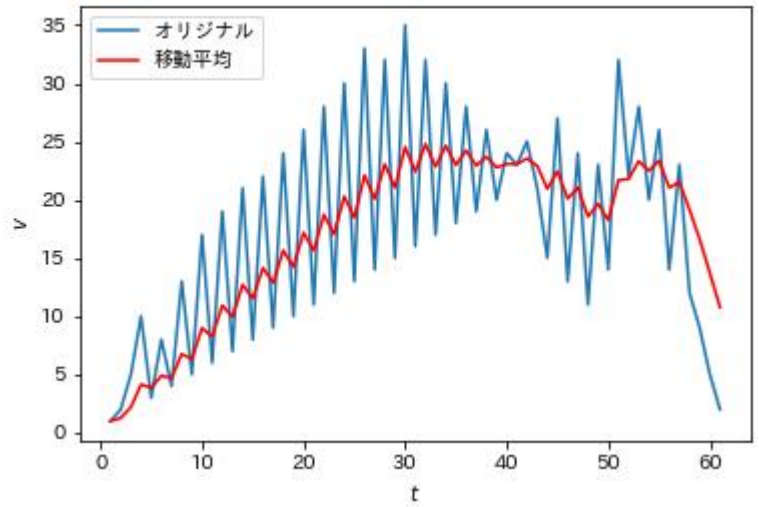
更新式をベクトル図で表すと、現在の勾配の向きに対して、今まで動いていた方向つまり慣性(モーメンタム)の方向を足し合わせた方向を最終的な方向 としています。



二つを足し合わせるため正負で振動していると打ち消しあって振動が小さくなるのです。

3.1.4 モーメンタム

なぜモーメンタムでSGDの振動を抑える ことができるのか?
⇒モーメンタムを考えることは指数平滑移動平均を考えることと同義です。



上のグラフは指数平滑移動平均を用いたグラフです。
青線に比べて赤線は振動を抑え、緩やかになっています。

つまり、指数移動平均は急な変化に動じないグラフになっている。

3.1.4 モーメンタム

指数平滑移動平均は以下の式で各点を求めます。

$$\nu_t = \beta\nu_{t-1} + (1-\beta)G$$

ここで ν_{t-1}, ν_tはそれぞれ前時刻での移動平均された後の値、Gは現時刻の値で、β は0から1の値を取るハイパーパラメータです。
つまり

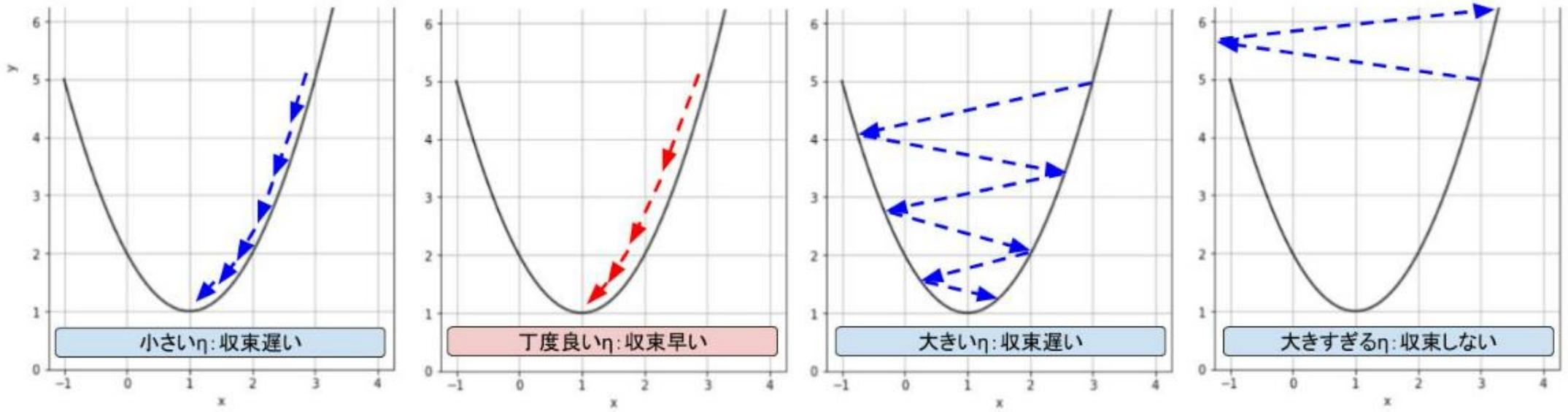
- **第1項 $\beta\nu_{t-1}$: 今までの G たちを蓄積した項**
- **第2項 $(1-\beta)G$: 現在の点を表す項**

またν_tについて漸化式を解くと、時間が経過するごとに与える影響は小さくなることがわかる

$$v_t = (1-\beta)(G_t + \beta G_{t-1} + \beta^2(G_{t-2})...$$

3.4 学習率の選定と制御

学習率の選び方は重要で下の図のように適当な学習率を選ばなければ、学習の速度や成否にかかわります。



そのため、学習率を学習を進めるにつれて変化させる工夫がなされています。

3.5.1 更新幅(学習率)の適応的変化 RMSProp

RMSPropはモーメンタムと同じくSGDの振動を抑えるという目的で作られたものです。ただRMSPropはgradの大きさに応じて学習率を調整するというものです。

$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \underline{\epsilon}\underline{\nabla E_n}$$

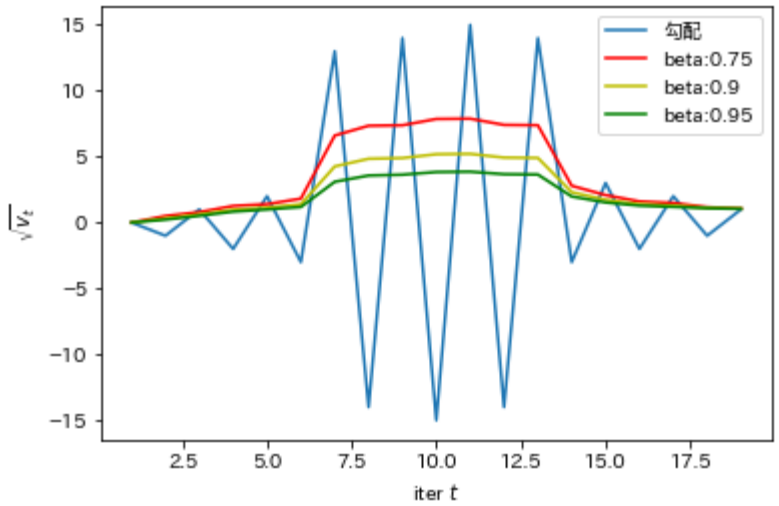
RMSProp　モーメンタム

まず第1式は、移動平均の式です。 ここで注意して欲しいのは G2G2 になっている こと です。これにより、モーメンタムのように振動を正負で打ち消しあうのではなく、振動している時の値はかなり大きくなります。

$$< g_i^2 >_t = \gamma < g_i^2 >_{t-1} + (1-\gamma) g_{t,i}^2$$

3.5.1 更新幅(学習率)の適応的変化 RMSProp

このグラフは元々の勾配(青色)に対するRMSPropの第1式の値たちです。勾配が急に変化するところでは第1式の値が大きくなっている ことがわかります。



$$\Delta w_{t,i} = -\frac{\epsilon}{\sqrt{<g_i^2>_t + \varepsilon}} g_{t,i}$$

第2式はパラメータを更新しています。元々の学習率を第1式の値で割ってあげるだけです。振動しているときは第1式が大きくなるので、振動しているときの分母が大きくなり学習率を小さくしたい、と言う目的が達成できます。

3.5.2 Adam

Adamの正体はモーメンタムとRMSPropの良いどこどり というものです。移動平均で振動を抑制するモーメンタム と 学習率を調整して振動を抑制するRMSProp を組み合わせているだけなのです。

$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \epsilon \nabla E_n$$

RMSProp　モーメンタム

Adam

$$m_{t,i} = \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$$

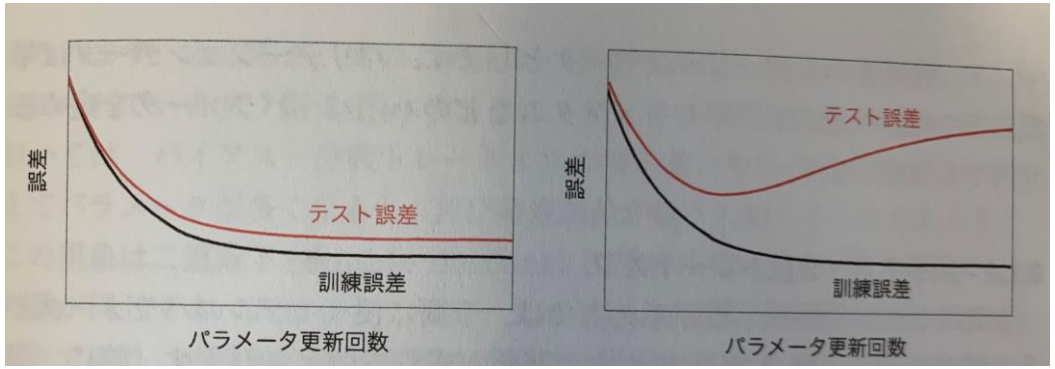
$$v_{t,i} = \beta_2 v_{t-1,i} + (1-\beta_2) g_{t,i}^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\Delta w_{t,i} = -\frac{\epsilon}{\sqrt{\hat{v_{t,i}}}+\varepsilon} \hat{m}_{t,i}$$

3.2.1 訓練誤差と汎化誤差



訓練誤差：訓練データに対する誤差
汎化誤差：サンプルの母集団に対する誤差

しかし、母集団は未知のデータのため真の分布関数やモデルがわからない。
ただ学習の成否を確かめるための指標が欲しい。

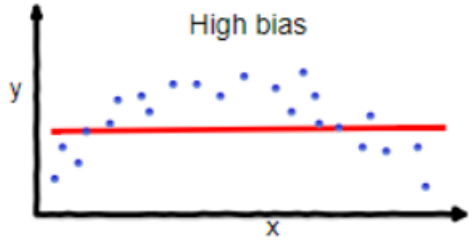
テスト誤差：サンプル集合からテスト用に取り出した集合に対する誤差
を用意する。

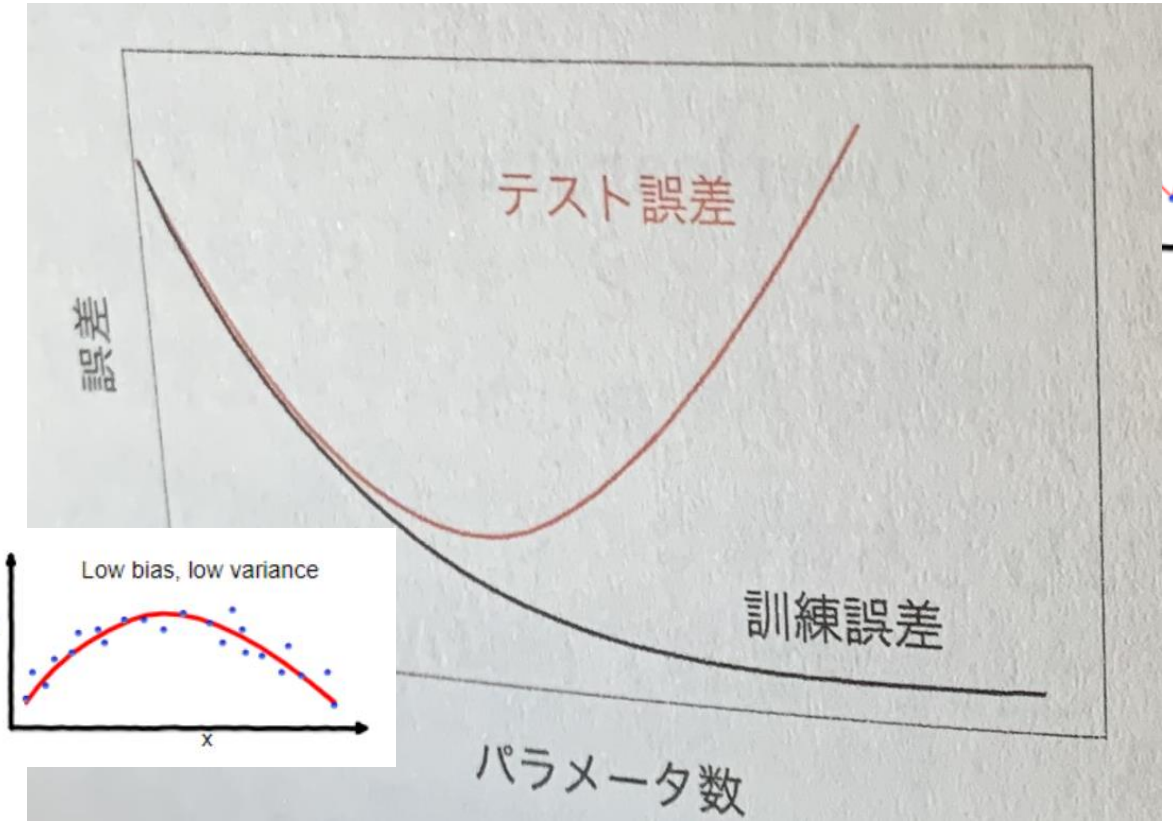
訓練誤差は、学習によってデータにモデルを近づけるため単調減少する。
一方で、テスト誤差ははモデルが訓練データに適合しすぎることで、外れ値なども学習してしまい（過剰適合、過学習）単純には減少しない。

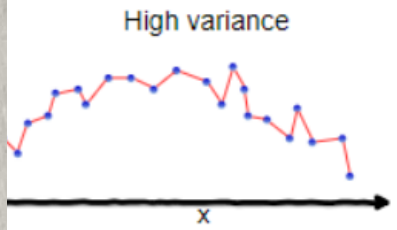
3.2.2 バイアス・分散トレードオフ

パラメータ数が少ないと、モデルの自由度が不足し説明できない。

パラメータ数が多いと、モデルの自由度が過剰となり外れ値などまで説明してしまう。







3.2.3 汎化と記憶

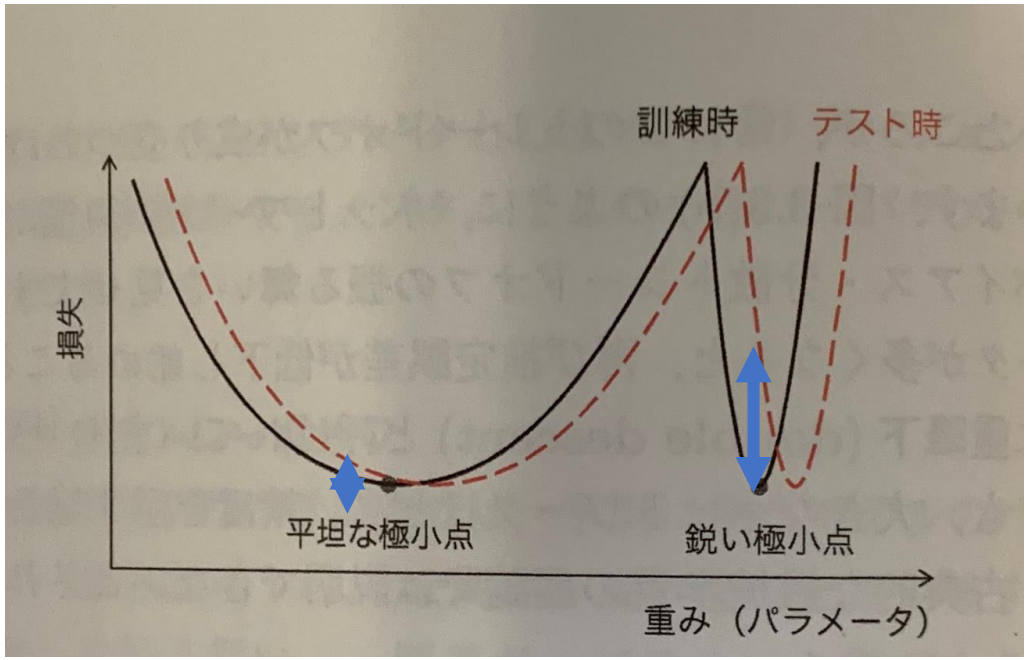
元のデータ　$[\text{入力 } x_i, \text{出力 } d_i]_{n=1\ldots N}$　を　$[\text{入力 } x_i, \text{出力 } d_j]_{n=1\ldots N}\ i \neq j$
と入力と出力をバラバラにする。

このバラバラにしたデータを学習することもできる、ただそれは全てのデータを記憶しただけ？

3.2.4 平坦な極小点

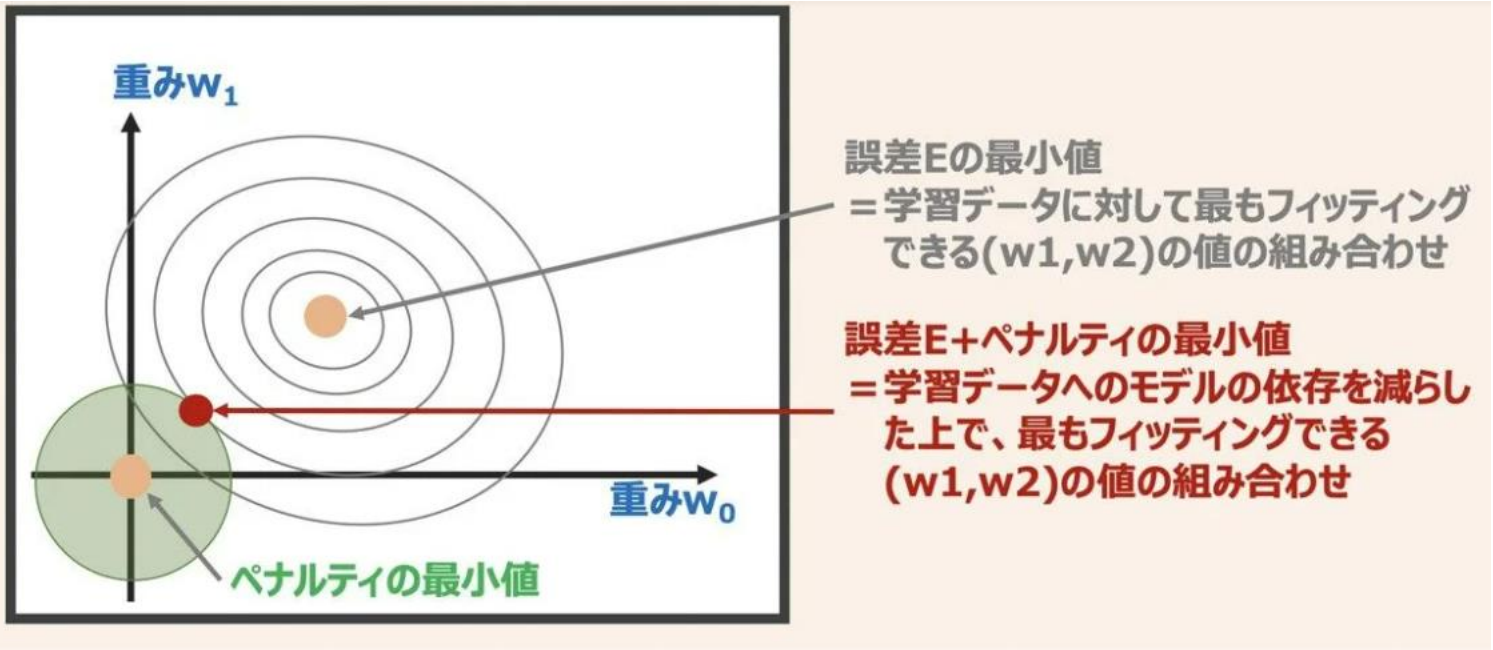
深厩学習で扱うデータや問題では,損失関数は一般に極小点を複数持つので、その中からなるべくよいものを見つけることが目標となる。

ここでの良い極小点は平坦な極小点とされます。
これは、下図の矢印のように、平坦な極小点の方が損失関数がずれた時の誤差が小さくなるからです。



3.3.1 正則化

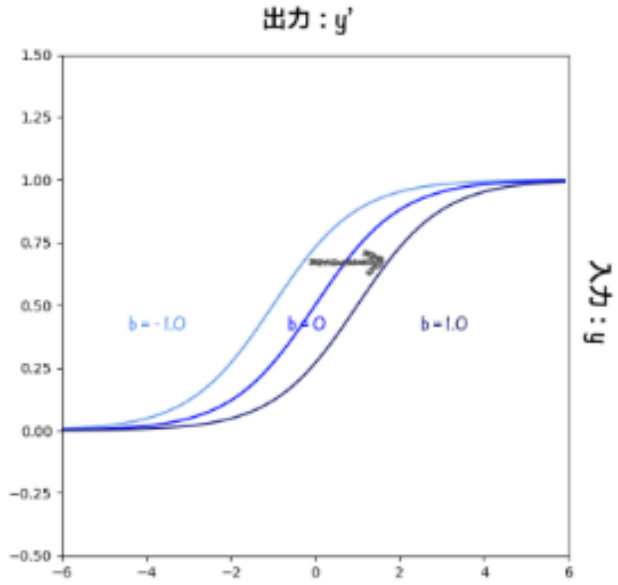
パラメータの変化に制限を与えることで、過剰適合を防ぐ考え。



$$E = \frac{1}{2}\sum (y_i - w_i x_i)^2 + R(w)$$

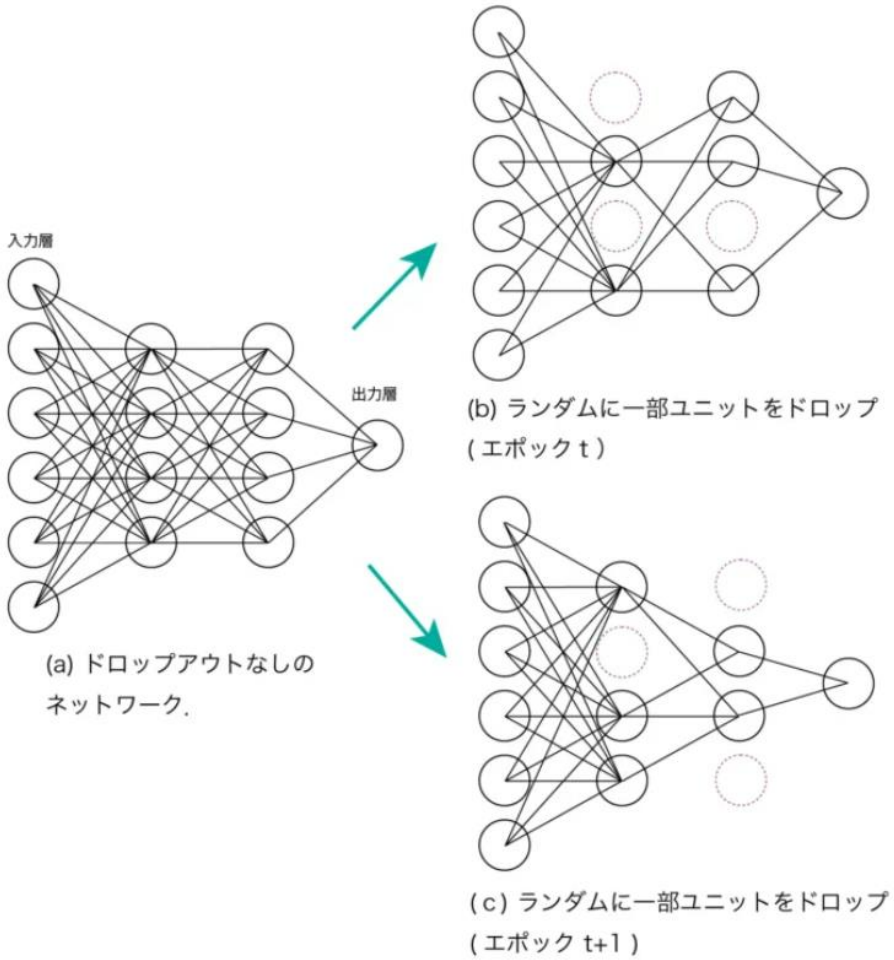
誤差項　　ペナルティ項

E：	誤差
R：	ペナルティ項
y_i：	正解値
x_i w_i：	予測値
x：	説明変数
w：	重み

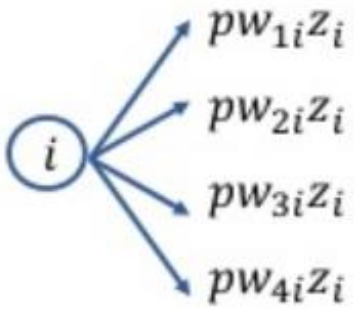


3.3.2 ドロップアウト

ドロップアウト とは、ランダムにノードを非活性にして学習させること。ノードを減らすことで、パラメータの自由度を下げて過学習を防ぐことができる。非活性になるノードはランダムのため学習のたびに異なる重みとなる。



推論時には、無効化を行った層出力をp倍にします。
推論時には非活性にしていたノードも考慮するため、ノード数が学習時に比べて1/p倍になるからです。



3.3.3 陰的正則化

正則化ではないが、過学習を防ぐ働きをもつもの。

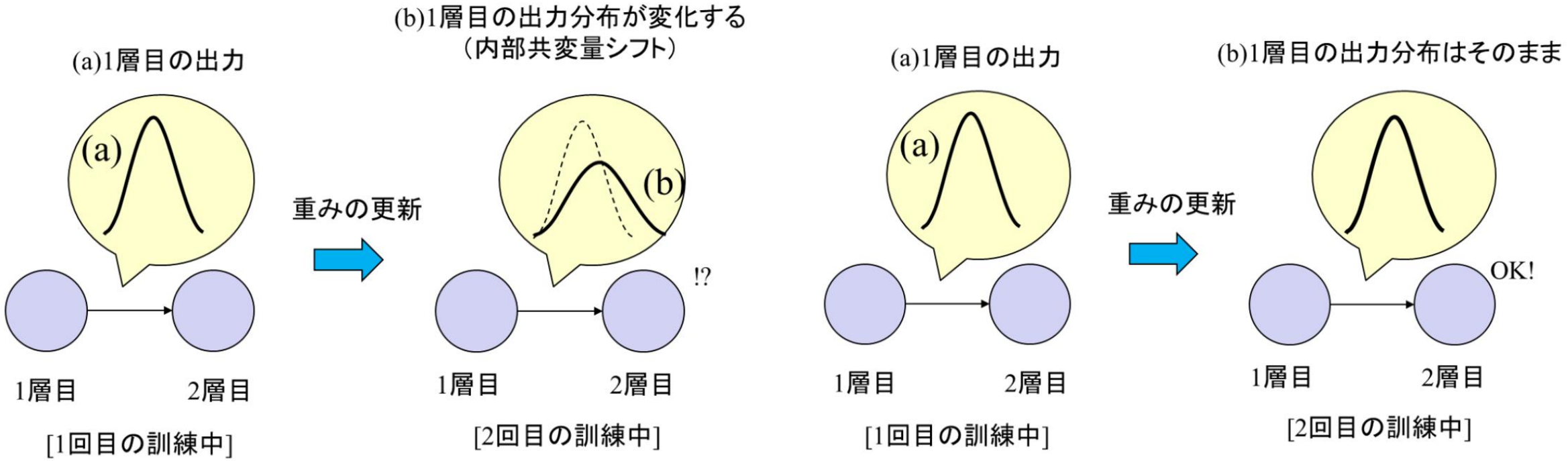
大きな学習率

ミニバッチのサイズを小さくすること

早期終了

3.6.1 層出力の正規化

入力がある統計的な分布に従うとすると、各層の出力も何らかの統計分析に従います。学習の進展とともに重みが変化すると、各層の出力が大きく変化してしまい学習に悪影響を与える。
これを解決するために正規化を行う。



3.6.2 入力の正規化

入力がの統計量に大きな違いがあると計算時に与える影響が大きくなり学習に悪影響を及ぼす可能性があります。これを解決するために正規化を行います。

年齢 = {"10", "15", "20", "25", "30"}

身長 = {"130", "165", "170", "173", "170"}

- W1とW2が1の場合(年齢と身長の重要度は同じ)

  10 × 1 + 130 × 1 = 140

  15 × 1 + 165 × 1 = 180

  20 × 1 + 170 × 1 = 190

  25 × 1 + 173 × 1 = 198

  30 × 1 + 170 × 1 = 200

- W1=1、W2=2の場合(年齢よりも身長を重要視する)

  10 × 1 + 130 × 2 = 270

  15 × 1 + 165 × 2 = 345

  20 × 1 + 170 × 2 = 360

  25 × 1 + 173 × 2 = 371

  30 × 1 + 170 × 1 = 370

- W1=2、W2=1の場合(身長よりも年齢を重要視する)

  10 × 2 + 130 × 1 = 150

  15 × 2 + 165 × 1 = 195

  20 × 2 + 170 × 1 = 210

  25 × 2 + 173 × 1 = 223

  30 × 2 + 170 × 1 = 230

3.6.2 入力の正規化

入力の正規化は以下の式で行います。

各成分からその成分の全学習データにわたる平均を求める。

$$平均\bar{x}_i = (1/N)\sum_{n=1}^{N} x_{n,i}$$

各成分からその平均を引き、変換後の平均が0になるようにします。

$$x_{n,i} \leftarrow x_{n,i} - \bar{x}_i$$

その後に分散を求めます。

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_{n,i} - \bar{x}_i)^2}$$

最後に分散で割ることで、各成分の平均が0、分散が１になり正規化される。

$$x_{n,i} \leftarrow \frac{x_{n,i} - \bar{x}_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

3.6.3 バッチ正規化

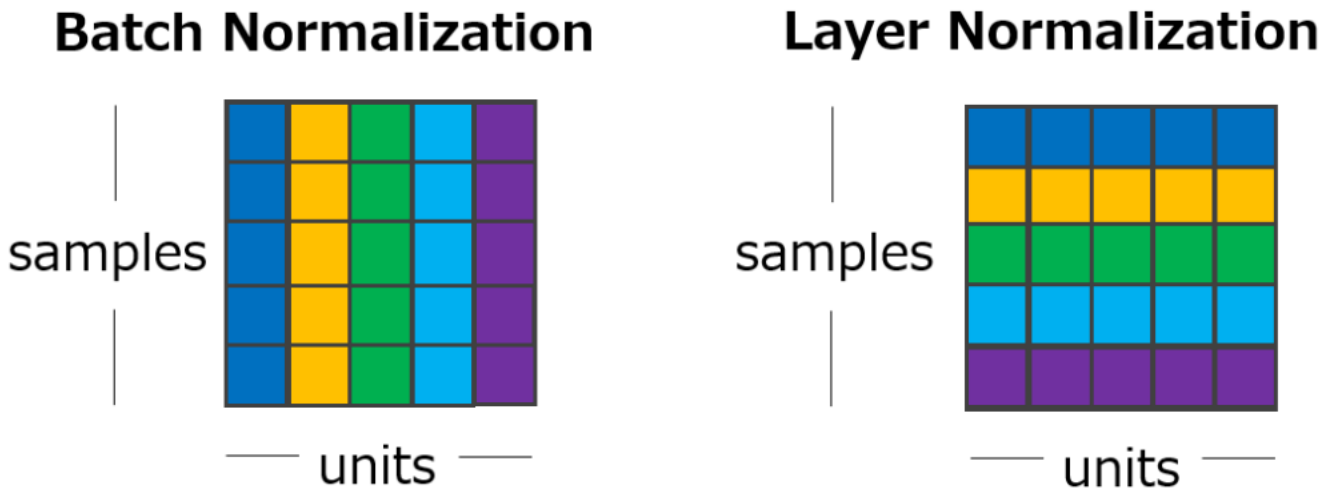
バッチ正規化ではミニバッチ内の全出力**u**に対する平均と分散を計算して用います。

$$u_j \leftarrow \gamma_j \frac{u_j - \mu_i}{\sqrt{\sigma_j^2 + \epsilon}} + \beta_j$$

この時のγ_jとβ_jはハイパーパラメータです。
正規化するだけでも学習は高速に収束するようになるが，それだけだと全出力**u**が同じ分布になりNNの表現力が落ちる。
ハイパーパラメータでスケールを変えて、シフトをさせることで，バッチ正規化層挿入前の変換を保つようにすることによりNNの表現力を保つ。

3.6.4 レイヤー正規化

バッチ正規化には二つの欠点があります。
一つ目は、メモリの関係などでミニバッチサイズを小さくすることがよくありますが、その場合、ミニバッチごとに平均・分散を計算すると推定値が不安定になってしまうということです。
二つ目は、再帰的ニューラルネットワークでは、各サンプルごとに文章の長さが違い、学習データよりも長い文章がテストデータにある場合、適用が簡単ではないということです。

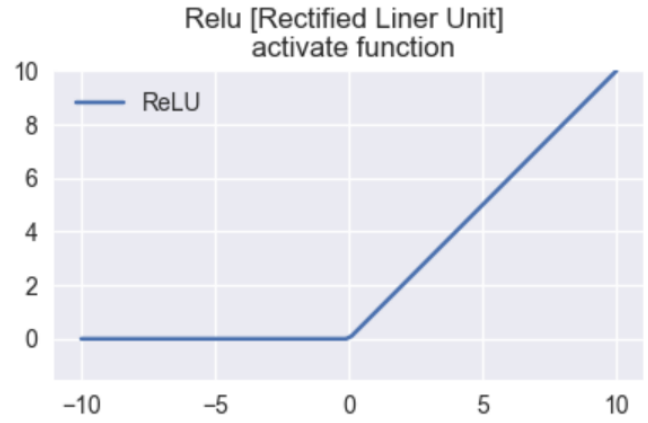


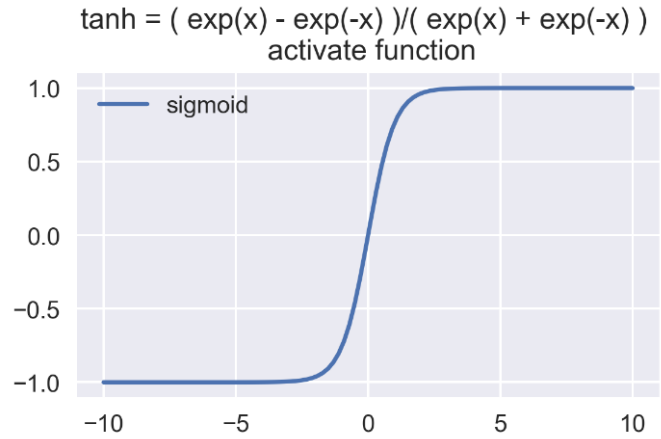
3.6.4 バッチ正規化の欠点

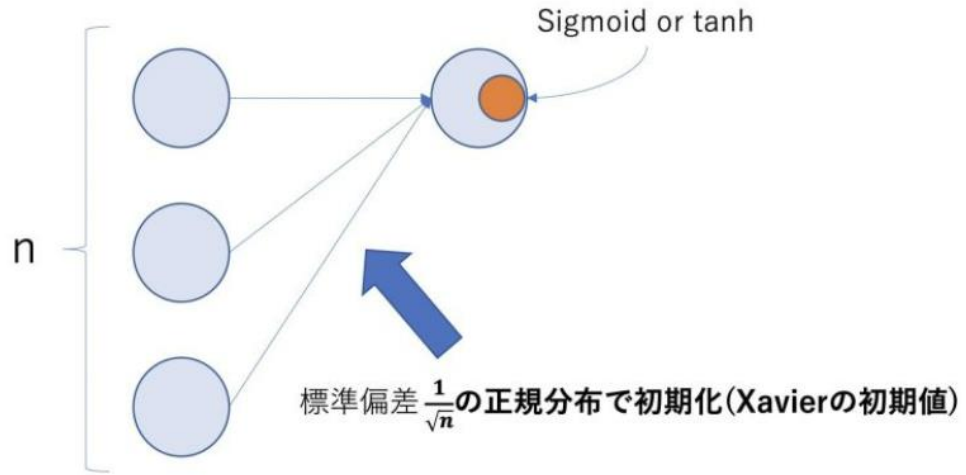
学習時にバッチ正規化を行うことで、統計量である平均と分散の正規化について学習を行っています。
一方で推定時には、学習で得られた正規化のパターメータを利用するためネットワークが訓練時と推論時で異なるふるまいをする？？

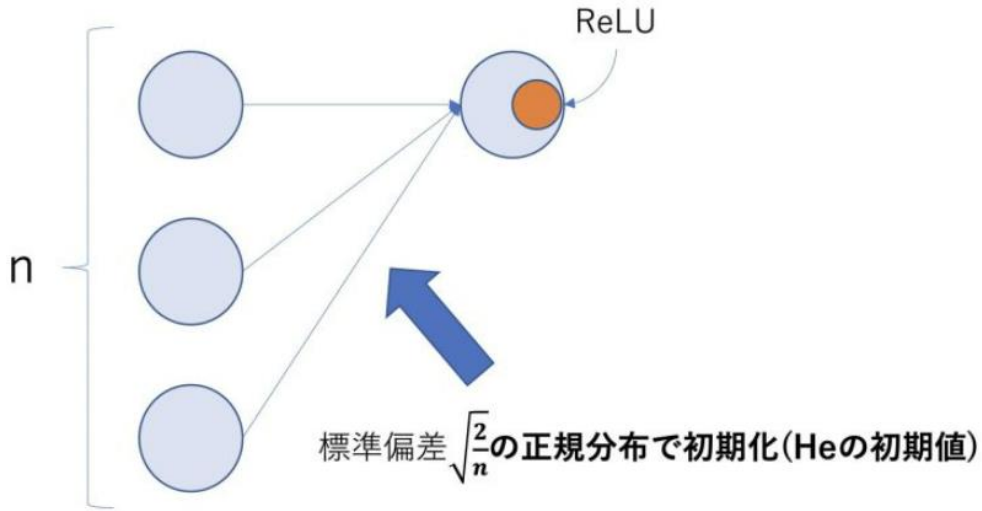
3.7　重みの初期化

重みの初期化の目的の一つはバッチ正規化と同様に統計量の違いを小さくするということです。
もう一つが活性化関数の適当な範囲に収めることです。









3.7　重みの初期化 Xavier法

ネットワークの損失関数をEとすると、パラメータ$s_k^{(i)}, w_{lk}^{(i)}$の微分は以下で書けます。

$$\frac{\partial E}{\partial s_k^{(i)}} = \left[f'(s^{(i)})\right]_k \sum_l w_{lk}^{(i+1)} \frac{\partial E}{\partial s_l^{(i+1)}}$$

$$\frac{\partial E}{\partial w_{lk}^{(i)}} = z_l^{(i)} \frac{\partial E}{\partial s_k^{(i)}}$$

各層での活性化関数の出力の分散$Var\left[z^{(i)}\right]$を同一に、また各層での誤差逆伝播の勾配の分散$Var\left[\frac{\partial E}{\partial s^{(i)}}\right]$を同一にすることを考えます。

活性化関数を一般のままにしてしまうと、解析が複雑になるため[1]では**「活性化関数は線形関数」**を仮定して議論しています**。**

つまり、$f'(x) = 1$を仮定します。Softsign関数もHyperbolic Tangentも原点付近では近似的にこの条件が成り立っています。また、**「初期化に関して平均0の分布を用いる」**ことも仮定しています。

一般に2つの独立な確率変数のX, Yの平均$E[X], E[Y]$、分散$Var[X], Var[Y]$とその和と積の分散$Var[X+Y], Var[XY]$の関係は以下です。

$$Var[X+Y] = Var[X] + Var[Y]$$

$$Var[XY] = Var[X]\,Var[Y] + Var[X]\,E[Y]^2 + Var[Y]\,E[X]^2$$

ここでX, Yの平均が0であると仮定すると、以下のように積の分散は分の積になります。

$$Var[XY] = Var[X]\,Var[Y]$$

これらの関係を用いると、$z^{(i)}$と$\frac{\partial E}{\partial s^{(i)}}$の分散$Var\left[z^{(i)}\right]$と$Var\left[\frac{\partial E}{\partial s^{(i)}}\right]$はそれぞれ以下の様に計算できます。

$$Var\left[z^{(i)}\right] = Var[x] \prod_{i'=0}^{i-1} n_{i'} Var\left[w^{(i')}\right]$$

$$Var\left[\frac{\partial E}{\partial s^{(i)}}\right] = Var\left[\frac{\partial E}{\partial s^{d}}\right] \prod_{i'=i}^{d-1} n_{i'+1} Var\left[w^{(i')}\right]$$

各層での分散が一定であるためには、全てのi, i'で以下が成り立つ必要があります。

$$\begin{cases} Var\left[z^{(i)}\right] = Var\left[z^{(i')}\right] \\ Var\left[\frac{\partial E}{\partial s^{(i)}}\right] = Var\left[\frac{\partial E}{\partial s^{(i')}}\right] \end{cases}$$

つまり、任意のiに対して以下が成り立てばよいです。

$$\begin{cases} n_i Var\left[w^{(i)}\right] = 1 \\ n_{i+1} Var\left[w^{(i)}\right] = 1 \end{cases}$$

この関係は、任意のiに対して$n_i = n_{i+1}$が成り立っていない限り解をもちません。そこでこれらの条件を出来る限り成り立つように、以下が成り立つよう要請してみます。

$$Var\left[w^{(i)}\right] = \frac{2}{n_i + n_{i+1}}$$

3.7　重みの初期化 He法

ここで考える非線形関数はReLU若しくはPReLUです。PReLUとはaを実数としたとき(通常は$0 \leq a < 1$)に以下のような関数です。$a = 0$の場合は特別にReLUと呼ばれます。

$$f(x) = \begin{cases} x \text{ for } x > 0 \\ ax \text{ for } x \leq 0 \end{cases}$$

[1]の初期化の導出では、線形な関数($a = 1$)を仮定してしていたため活性化関数の出力の平均は0として良かったですが、ReLUやPReLUの場合は非対称であるため、そうではありません。

平均がゼロでないためXavierの初期化の章で解説したように、確率変数の積の分散は分散の積になりません($Var[XY] \neq Var[X]Var[Y]$)。この点を修正したものがHeの初期化です。

具体的には以下のように変更されます。w自体は平均0の対称な分布で初期化されるため変更はありませんが、活性化関数を通した後のzに関しては平均は0ではありません。つまり、順伝播の分散はXavierの初期化の章での計算から以下となります。

$$\begin{aligned} Var\left[s^{(i)}\right] &= n_i Var\left[w^{(i)} z^{(i)}\right] \\ &= n_i Var\left[w^{(i)}\right] E\left[z^{(i)\,2}\right] \\ &= \frac{1}{2}(1+a^2) n_i Var\left[w^{(i)}\right] Var\left[s^{(i-1)}\right] \end{aligned}$$

2段目の等式が変更点です。3段目の等式でPReLUの関数形の情報を用いています。

逆伝播の勾配の分散も同様に以下になります。

$$\begin{aligned} Var\left[\frac{\partial E}{\partial z^{(i)}}\right] &= n_{i+1} Var\left[w^{(i)} \frac{\partial E}{\partial s^{(i)}}\right] \\ &= n_{i+1} Var\left[w^{(i)}\right] Var\left[\frac{\partial E}{\partial s^{(i)}}\right] \\ &= \frac{1}{2}(1+a^2) n_{i+1} Var\left[w^{(i)}\right] Var\left[\frac{\partial E}{\partial z^{(i+1)}}\right] \end{aligned}$$

2段目の等式で$w^{(i)}$と$\frac{\partial E}{\partial s^{(i)}}$が独立という仮定を用いています。3段目の等式でPReLUの関数形の情報を用いています。

以上から任意のiで以下が成り立ちます。

$$Var\left[s^{(i)}\right] = Var[x] \prod_{i'=0}^{i-1} \frac{1}{2}(1+a^2) n_{i'} Var\left[w^{(i')}\right]$$

$$Var\left[\frac{\partial E}{\partial z^{(i)}}\right] = Var\left[\frac{\partial E}{\partial s^{d}}\right] \prod_{i'=i}^{d-1} \frac{1}{2}(1+a^2) n_{i'+1} Var\left[w^{(i')}\right]$$

Xavierの初期化の時と同様に、分散が一定であるためには以下が成り立てばよいです。

$$\begin{cases} \frac{1}{2}(1+a^2) n_i Var\left[w^{(i)}\right] = 1 \\ \frac{1}{2}(1+a^2) n_{i+1} Var\left[w^{(i)}\right] = 1 \end{cases}$$

この関係は任意のiに対して$n_i = n_{i+1}$が成り立っていない限り解をもちません。

Xavierの初期化の場合は、そこでこれらの条件を出来る限り成り立つように以下が成り立つようにしましたが、[1]ではどちらか一方が成り立つように要請します。つまり以下のどちらかを満たすようにします。

$$\begin{cases} Var\left[w^{(i)}\right] = \frac{2}{(1+a^2) n_i} \text{ for all } i \\ Var\left[w^{(i)}\right] = \frac{2}{(1+a^2) n_{i+1}} \text{ for all } i \end{cases}$$

このような要請を満たす初期化が[2]で提案されているHeの初期化です。