6.2 重みの初期値

高柳海斗(リュカ)

重みの初期値

ニューラルネットワークの性能は初期値によっても左右される

今まで使っていた初期値

```
0.01 * np.random.randn(10, 100)
```

- よい初期値の考察
  - 重みが大きくなると過学習を起こす傾向にある
  - 小さい値からスタートするのがよいのでは?

重みの初期値を0にする？

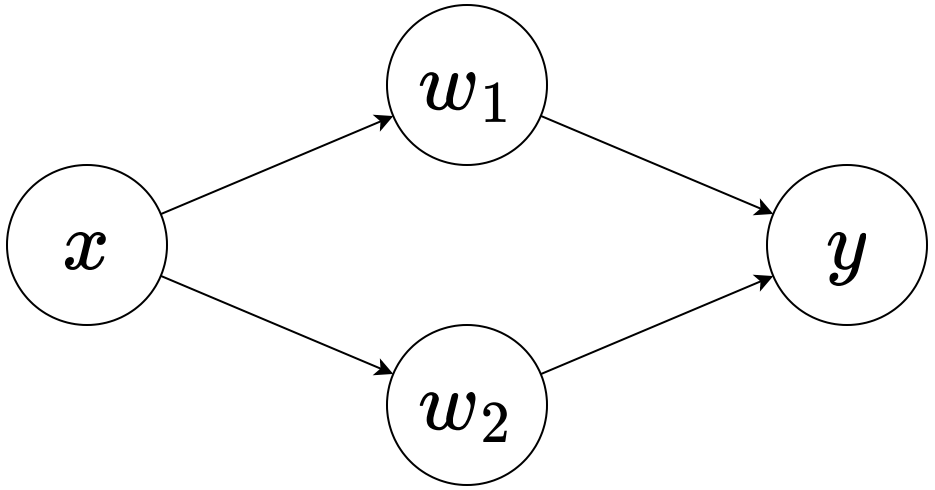
小さい値からスタートするのがよいならすべて0にしてしまえば良いのでは？→ 実は誤り

$$w_1 = \sigma(a_1 x + b_1)$$
$$w_2 = \sigma(a_2 x + b_2)$$
$$y = \sigma(a_3 w_1 + a_4 w_2 + b_3)$$

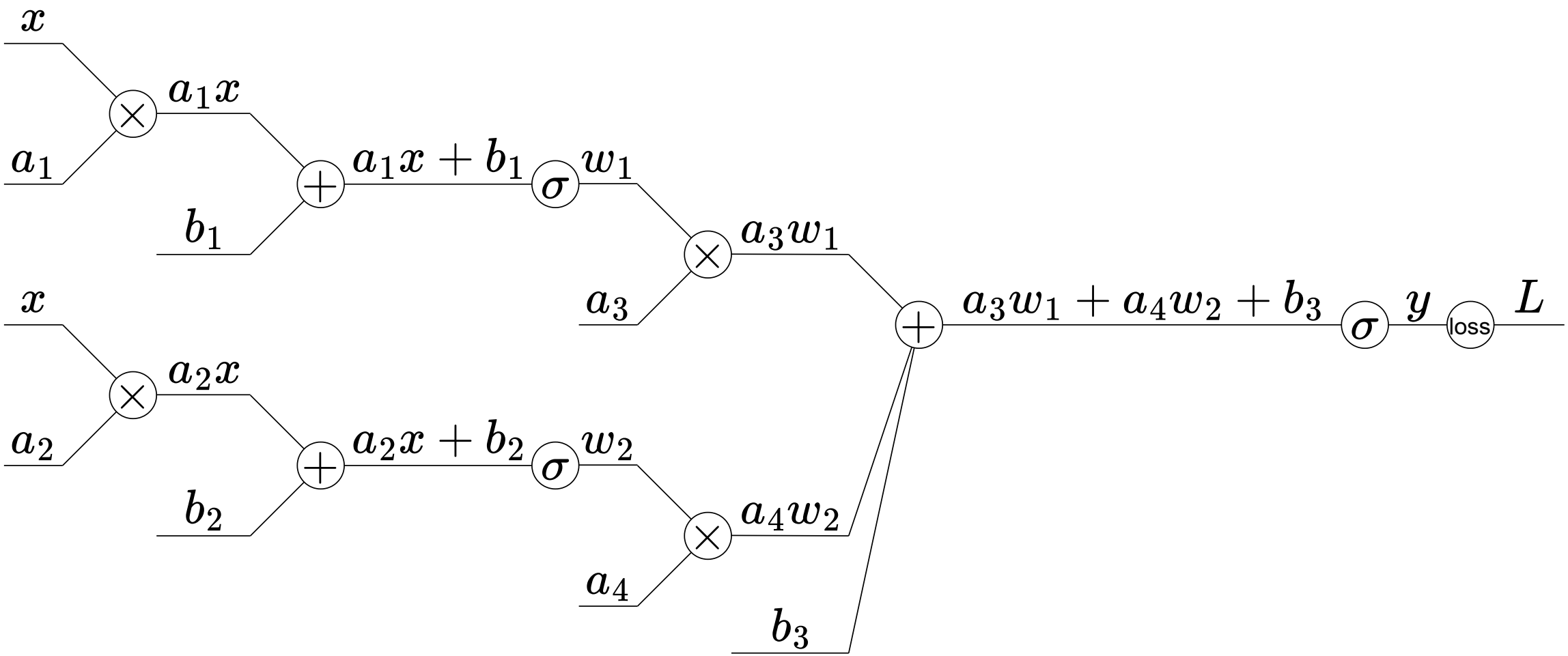
パラメータの初期値を0とする

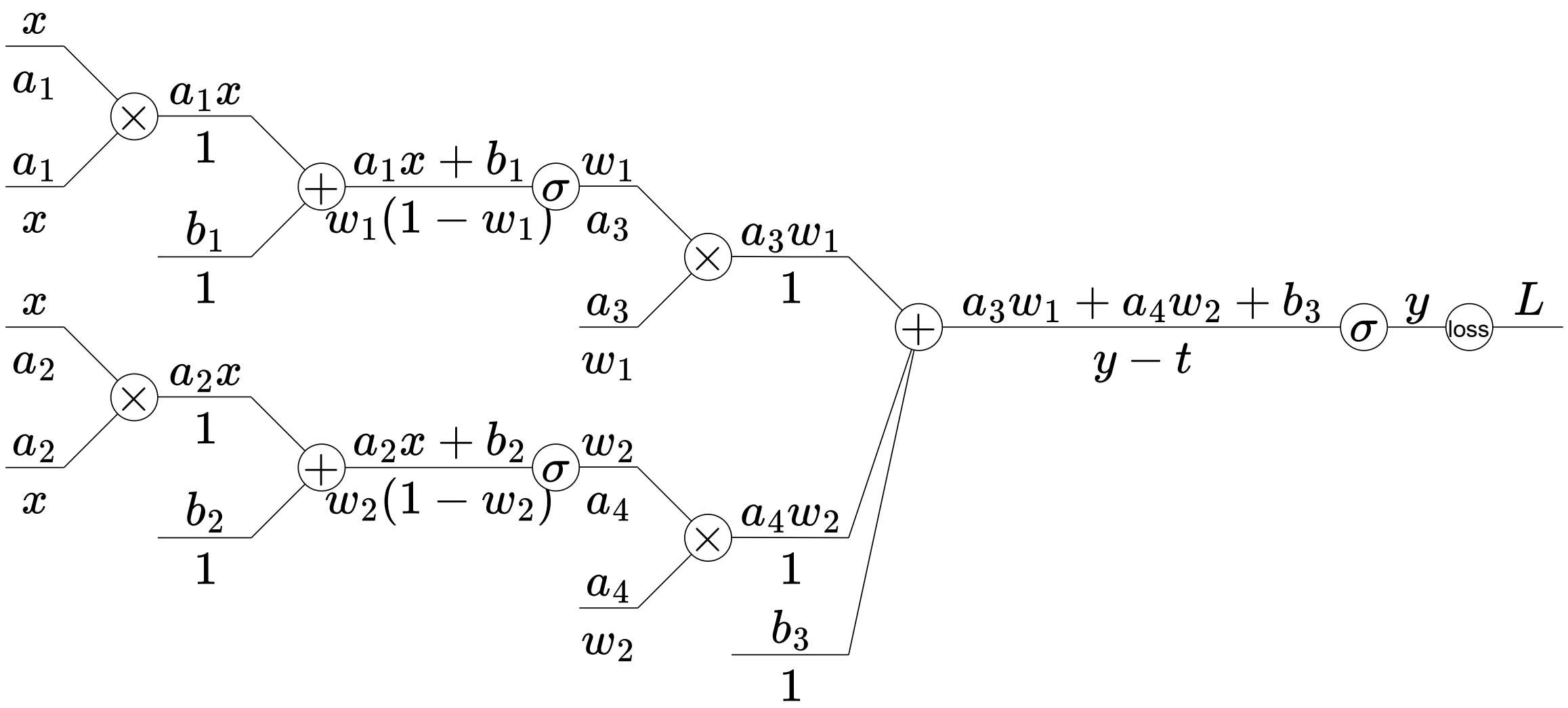
対称性より

$$\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial a_2}, \frac{\partial L}{\partial a_3} = \frac{\partial L}{\partial a_4}, \frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial b_2}$$



単純なネットワーク


x
a_1
a_1x
b_1
$a_1x + b_1$
σ
w_1
a_3
a_3w_1
x
a_2
a_2x
b_2
$a_2x + b_2$
σ
w_2
a_4
a_4w_2
b_3
$a_3w_1 + a_4w_2 + b_3$
σ
y
loss
L


x
a_1
a_1
x
$a_1 x$
1
b_1
1
$a_1 x + b_1$
$w_1(1 - w_1)$
σ
w_1
a_3
$a_3 w_1$
1
a_3
w_1
x
a_2
a_2
x
$a_2 x$
1
b_2
1
$a_2 x + b_2$
$w_2(1 - w_2)$
σ
w_2
a_4
$a_4 w_2$
1
a_4
w_2
b_3
1
$a_3 w_1 + a_4 w_2 + b_3$
$y - t$
σ
y
loss
L

正解をtとして勾配を計算すると

$$\frac{\partial L}{\partial a_1} = xw_1(1-w_1)a_3(y-t) \qquad \frac{\partial L}{\partial a_4} = w_2(y-t)$$

$$\frac{\partial L}{\partial a_2} = xw_2(1-w_2)a_4(y-t) \qquad \frac{\partial L}{\partial b_1} = w_1(1-w_1)a_3(y-t)$$

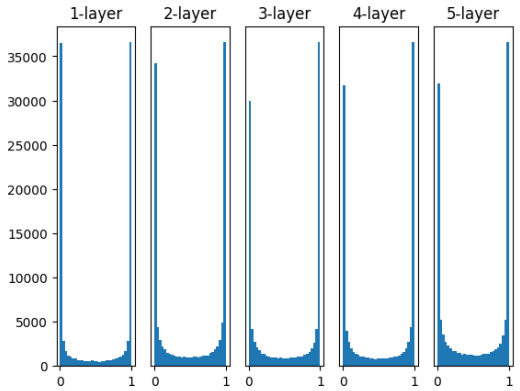
$$\frac{\partial L}{\partial a_3} = w_1(y-t) \qquad \frac{\partial L}{\partial b_2} = w_2(1-w_2)a_4(y-t)$$

初期値が$a_1 = a_2, a_3 = a_4, b_1 = b_2$を満たしていると対称性を保存してしまうことがわかる

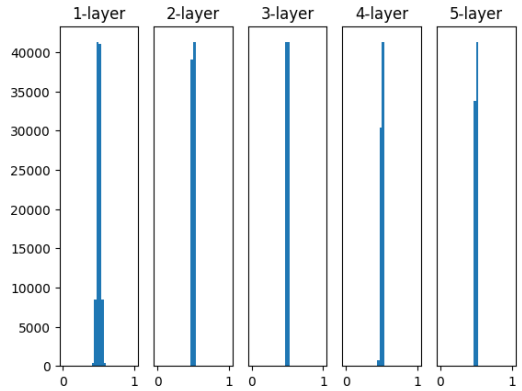
ゆえに初期値にはある程度のばらつきが必要

隠れ層のアクティベーション分布

隠れ層のアクティベーション(活性化関数の後の出力データ)を観察することで初期値の良さを評価できる



標準偏差1の正規分布



標準偏差0.01の正規分布

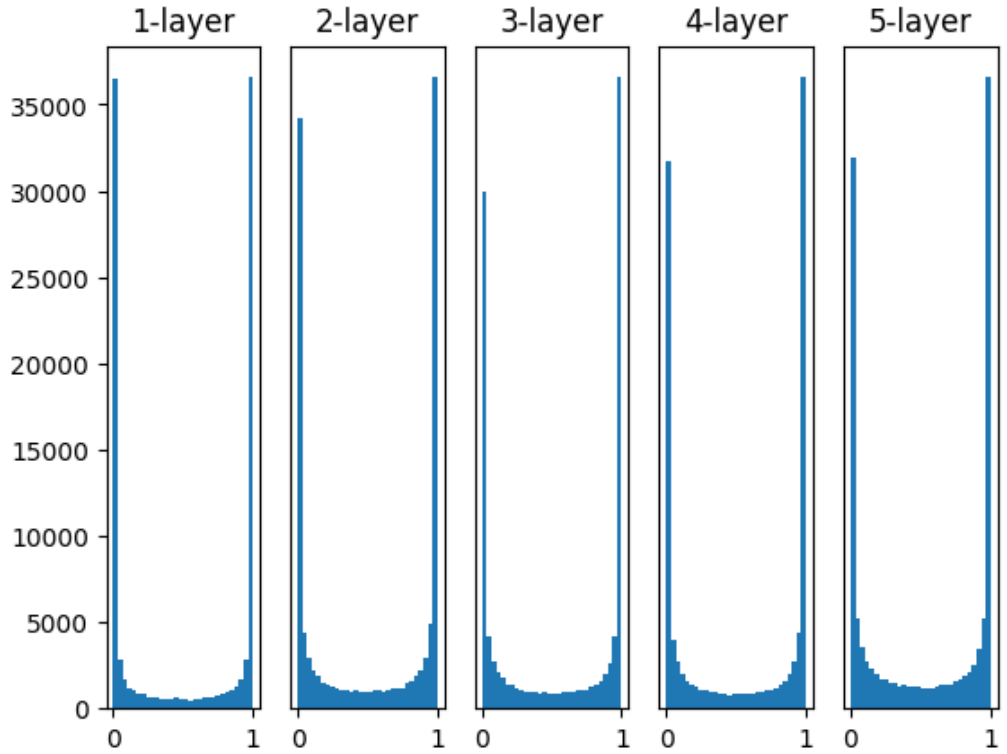
ソースコード: `\6.2src\weight_init_activation_histogram.ipynb`

勾配消失 (gradient vanishing)

右の図を見ると各層のアクティベーションは0と1に偏った分布になっていることがわかる

シグモイド関数の両端では微分係数が0に近くなっており, 逆伝播での勾配の値が小さくなってしまうことから学習に時間がかかる

これを**勾配消失**という

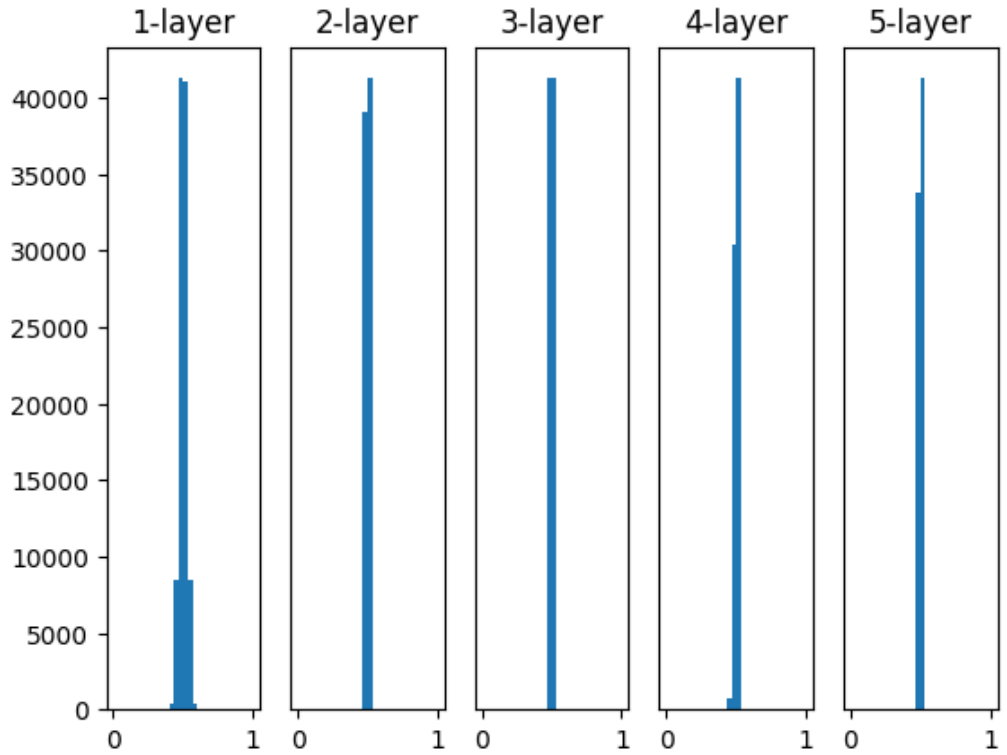


標準偏差1の正規分布

表現力の制限

一方右の図を見ると0.5付近に集中していることが分かる

複数のニューロンがほとんど同じ値を出力するとすれば、(例で見たように)1個のニューロンでもほぼ同じことを表現できるため、ネットワークの表現力が落ちてしまう問題が起こる

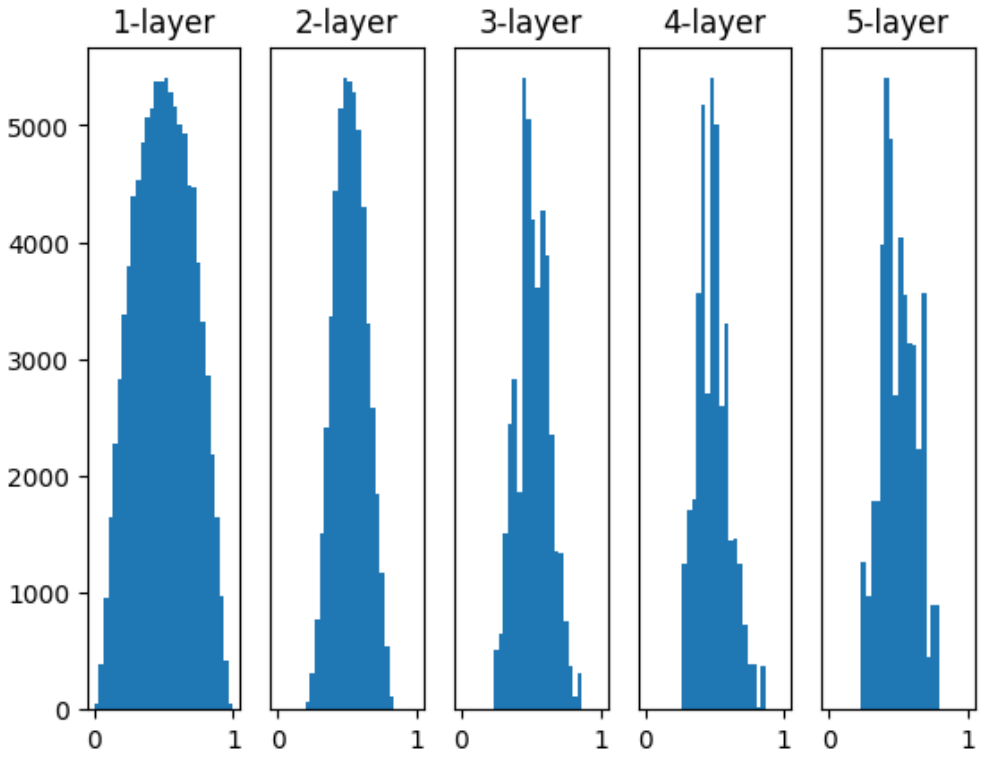


標準偏差0.01の正規分布

Xavierの初期値

Xavier Glorotらの論文で推奨されている初期値「Xavierの初期値」

- 分布:前層のノードの個数をnとして、標準偏差$1/\sqrt{n}$の正規分布
- 仮定:活性化関数は線形
- 適用:活性化関数にシグモイドやtanhを使っているとき

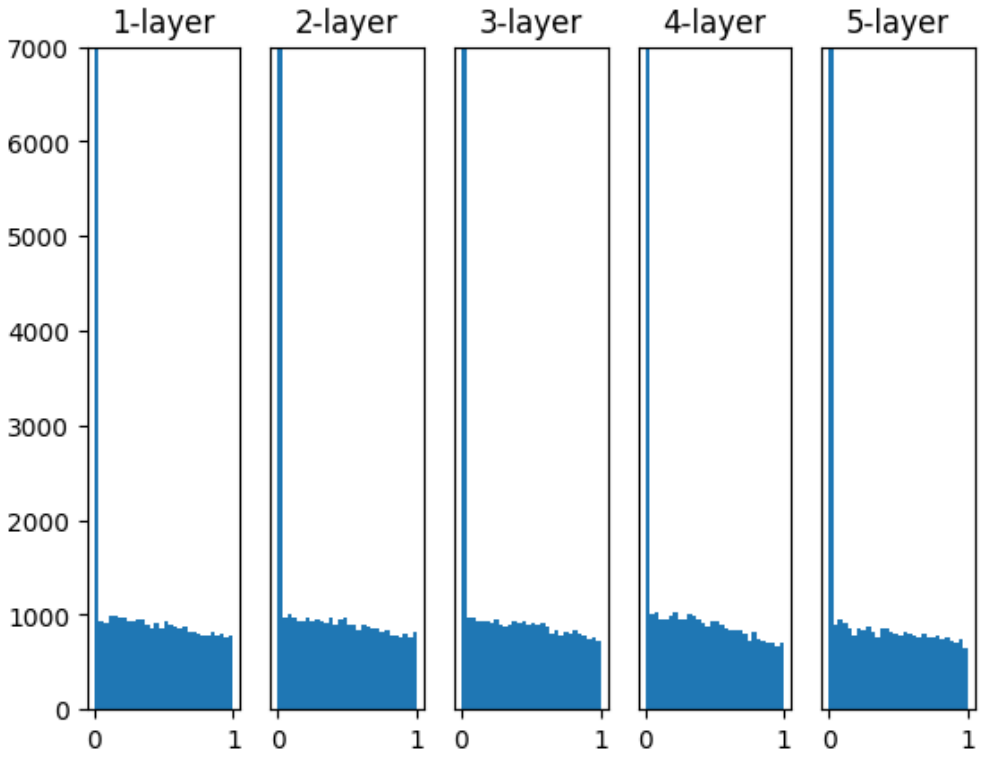


Xavierの初期値

Heの初期値

Kaiming He(何恺明)らが提案した「Heの初期値」

- 分布:前層のノードの個数をnとして、標準偏差$2/\sqrt{n}$の正規分布
- 仮定:活性化関数はReLU
- 適用:活性化関数にReLUやその派生を使っているとき



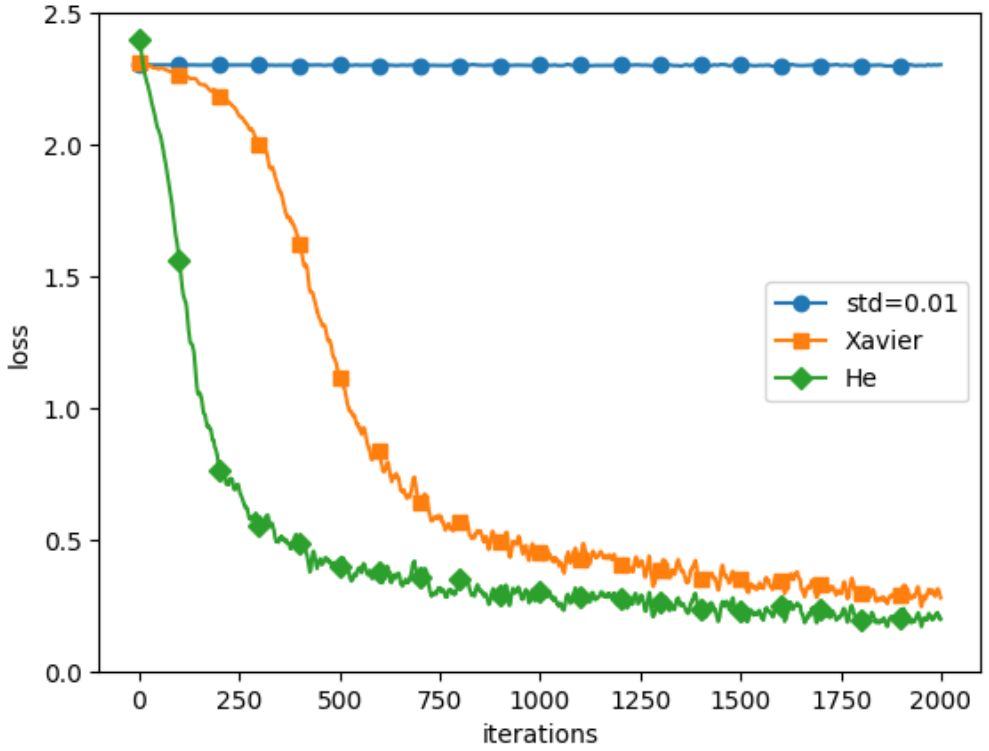
Heの初期値

MNISTによる重み初期値の比較

ソースコード:

`\6.2src\weight_init_compare.ipynb`

図を見るとHeの初期値が最も速く収束していて、Xavierはそこそこ、0.01の正規分布にいたってはほとんど学習できていないことがわかる



初期値ごとのLossの推移

6.3 Batch Normalization

by リュカ

Batch Normalization

重みの初期値を適切に設定すれば、各層のアクティベーションの分布は適度な広がりを持ち、学習がスムーズに行えることが分かった

では各層で"強制的に"アクティベーションの分布を調整させたらどうだろうか

このようなアイデアをもとにする手法が Batch Normalizationである

Batch Normalizationのアルゴリズム

Batch Normalizationの利点

- 学習が速くなる(学習係数を大きくできる)
- 初期値にそれほど依存しない
- 過学習を抑制する(Dropoutなどの必要性を減らす)

ソースコード: `\6.2src\batch_norm_test.ipynb`