



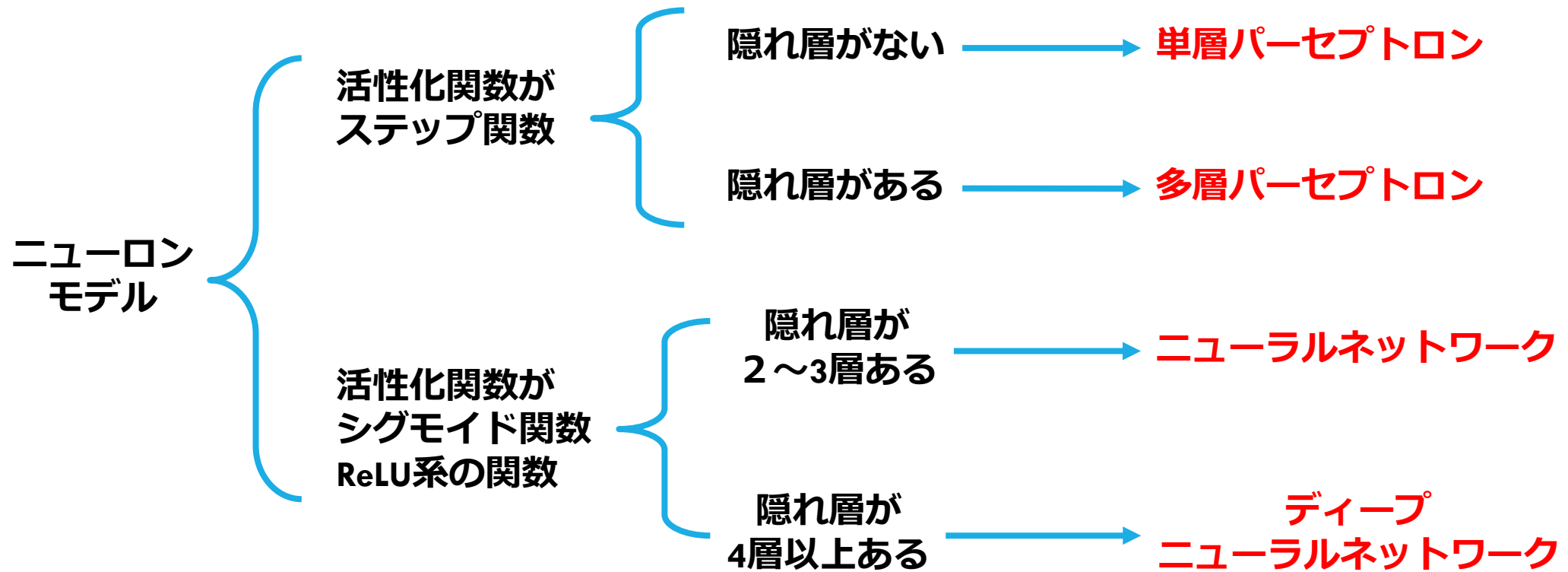
深層学習入門

#4 3層ニューラルネットワークと行列計算

たくのろじい / takunology

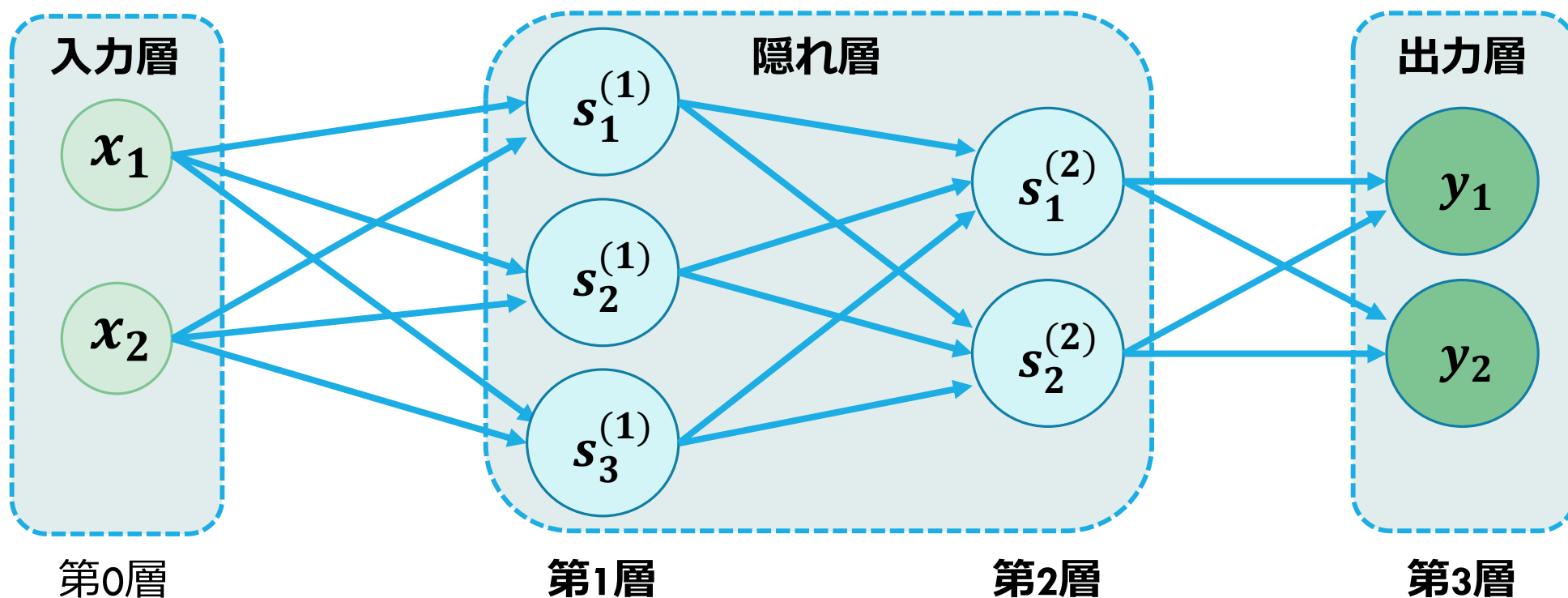
パーセプトロンとニューラルネットワーク

- パーセプトロンとニューラルネットワークの違いは？



3層ニューラルネットワーク

- 入力層、**隠れ層×2**、出力層の計4層で構成される（ニューロンの数は任意）
- 入力層は活性化関数をもたないため、第0層とカウントする（入力層は数えない）
- ニューラルネットワークの計算には**行列計算**を用いる
- 上付き文字は**重み（隠れ層の数）**を意味する



行列の積

- 行と列の積を加算した結果が行列の積になる
- 行列の積は可換でないことに注意
- $[2 \times 2]$ と $[2 \times 2]$ の積は $[2 \times 2]$ として得られる（緑文字の要素数が解の要素数になる）

1行1列 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 + 4 & 3 + 2 \\ 12 + 8 & 9 + 4 \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$

1行2列 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 + 4 & 3 + 2 \\ 12 + 8 & 9 + 4 \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$

2行1列 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 + 4 & 3 + 2 \\ 12 + 8 & 9 + 4 \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$

2行2列 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 + 4 & 3 + 2 \\ 12 + 8 & 9 + 4 \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$

行列の型

- 行列の行数と列数は型として表現できる

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ 2行2列} \rightarrow [2 \times 2] \text{ 型}$$

$$\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \text{ 3行2列} \rightarrow [3 \times 2] \text{ 型}$$

- 行列の積は“列 - 行間”で共通の要素数がないと計算できない

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} \quad \begin{matrix} [2 \times 2] \text{ と } [2 \times 2] \text{ で } 2 \text{ が共通なので計算可} \\ \text{積は } [2 \times 2] \text{ 型になる} \end{matrix}$$

$$(a \ b) \begin{pmatrix} c & d & e \\ f & g & h \end{pmatrix} \quad \begin{matrix} [1 \times 2] \text{ と } [2 \times 3] \text{ で } 2 \text{ が共通なので計算可} \\ \text{積は } [1 \times 3] \text{ 型になる} \end{matrix}$$

$$\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \begin{pmatrix} g & h & i \\ j & f & l \end{pmatrix} \quad \begin{matrix} [3 \times 2] \text{ と } [2 \times 3] \text{ で } 2 \text{ が共通なので計算可} \\ \text{積は } [3 \times 3] \text{ 型になる} \end{matrix}$$

ハンズオン

- C# (.NET Core 3.1) で行列の積の計算プログラムを実装してみる
- 右の図は出力例

入力ロジックのポイント

1. 行と列の数を入力
2. 配列要素を for 文で回す
3. 行×列の数だけ要素を入力

出力ロジックのポイント

1. 行×列の数だけ for 文を回す
2. 要素を1つずつ表示
3. 要素は[行, 列]の順

行列の積ロジックのポイント

1. 共通要素数を新たな変数 k として3重 for 文を回す
2. 行列AとBの積を k 回 (共通要素の数だけ)加算していく
3. 保持した要素を表示

行列の積プログラムはGithubにアップしています

<https://github.com/takunology/DeepLearning/tree/master/MatrixCalc>

```
F:\GitHub\DeepLearning\MatrixCalc
Matrix Calc.
行列Aのパラメータ
行数: 2
列数: 2
0行0列の要素: 1
0行1列の要素: 2
1行0列の要素: 3
1行1列の要素: 4

行列Bのパラメータ
行数: 2
列数: 2
0行0列の要素: 4
0行1列の要素: 3
1行0列の要素: 2
1行1列の要素: 1

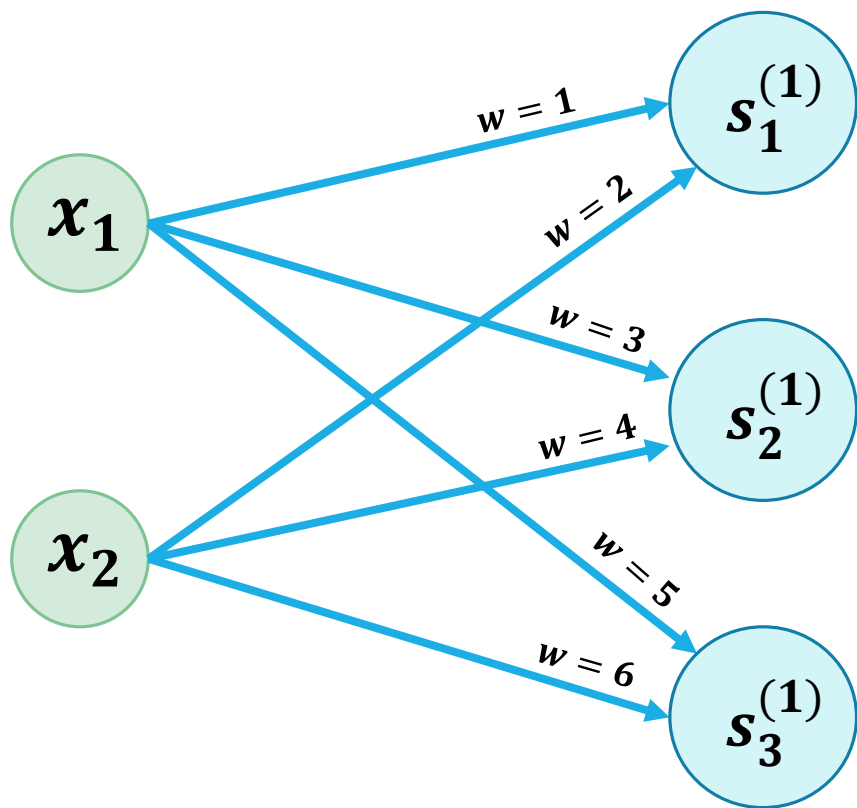
A =
| 1 2 |
| 3 4 |

B =
| 4 3 |
| 2 1 |

AB =
| 8 5 |
| 20 13 |
```

ニューラルネットワークと行列の積

- 入力値と重みの積を加算したもの → 行列の積の計算と同じ

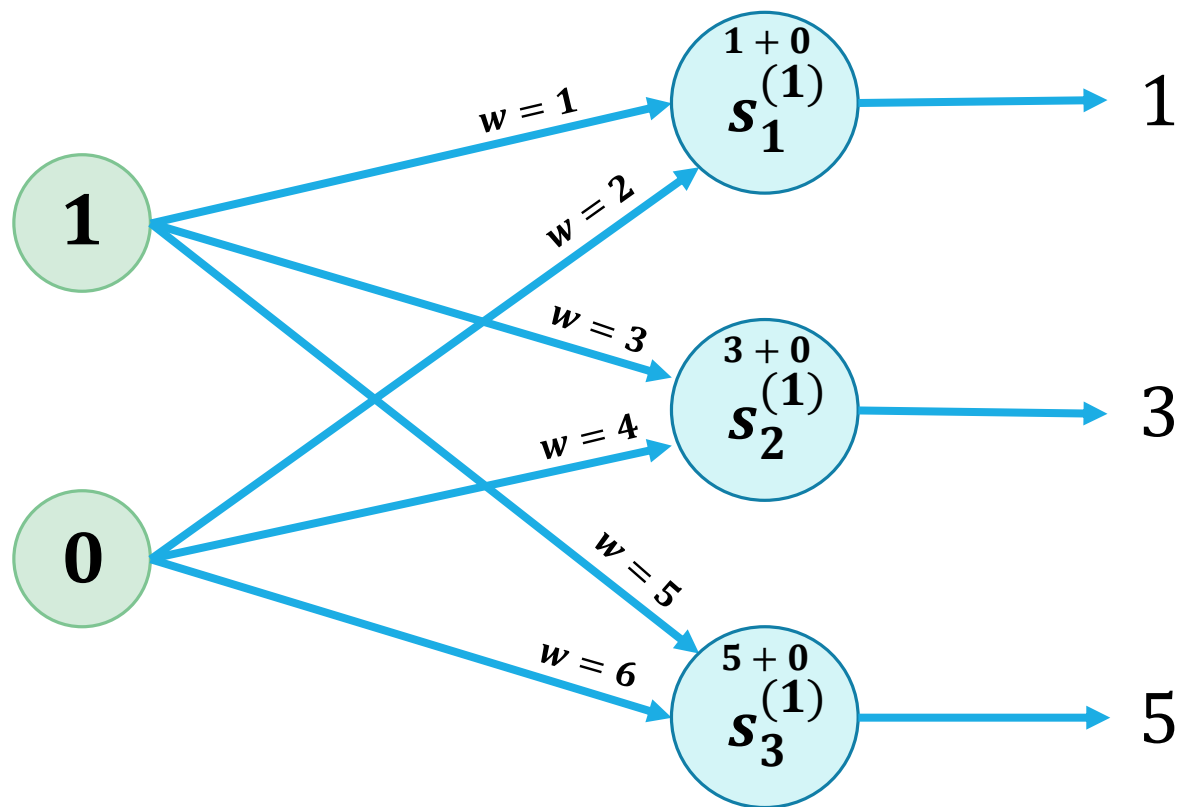


$x = (x_1 \quad x_2)$ 入力数

$w = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ 列 : 入力数
行 : ニューロン数

3層ニューラルネットワークと行列の積 #1

- 入力が (1, 0) だった場合の計算例
- 行列の積と出力結果が一致する



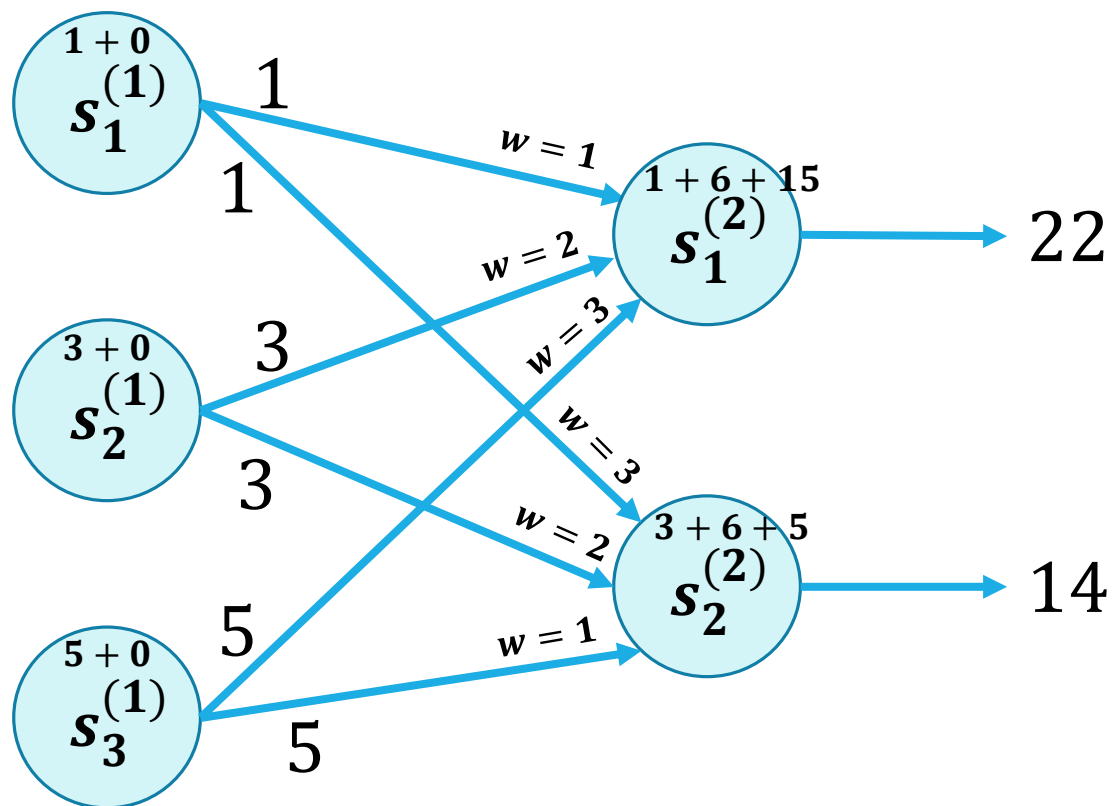
入力と重みの行列積
第1層目

$$\begin{aligned} & (1 \ 0) \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \\ &= \begin{pmatrix} 1+0 \\ 3+0 \\ 5+0 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} \end{aligned}$$

※解は[1×3]型ですが都合上、転置します

3層ニューラルネットワークと行列の積 #2

- 入力が (1, 0) だった場合の計算例
- もう1層通過させてみると...

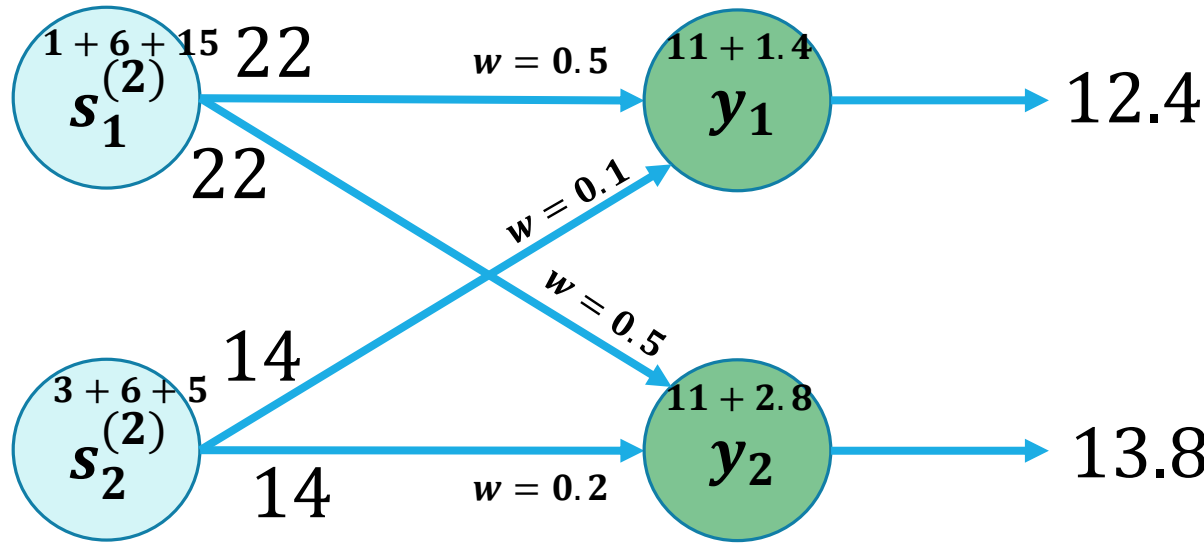


入力と重みの行列積
第2層目

$$(1 \ 3 \ 5) \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 + 6 + 15 \\ 3 + 6 + 5 \end{pmatrix} = \begin{pmatrix} 22 \\ 14 \end{pmatrix}$$

3層ニューラルネットワークと行列の積 #3

- 入力が (1, 0) だった場合の計算例
- 出力層を通すと...



入力と重みの行列積
第3層目

$$(22 \ 14) \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.2 \end{pmatrix} = \begin{pmatrix} 11 + 1.4 \\ 11 + 2.8 \end{pmatrix} = \begin{pmatrix} 12.4 \\ 13.8 \end{pmatrix}$$

3層ニューラルネットワーク全体

- ニューラルネットワークの図は行列を使って簡潔にできる

$$(1 \ 0) \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.2 \end{pmatrix} = \begin{pmatrix} 12.4 \\ 13.8 \end{pmatrix}$$

