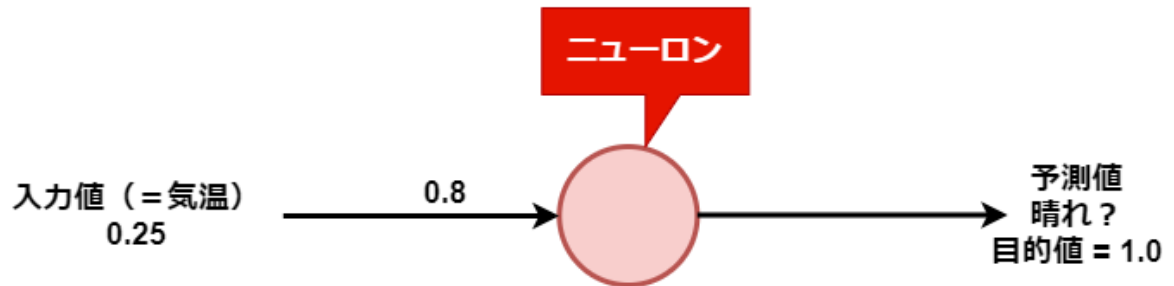


第4章 実習2

問題 1

▶ 下記のニューラルネットワークを実装するプログラムをコメントに従って実装してください。

◎実装するニューラルネットワーク



```

In [1]: ...,
        空のニューラルネットワーク neural_network の定義
        関数名 : neural_network
        引数 : input -> 入力、weight -> 重み
        処理 : 入力と重みを掛けて返す
        戻り値 : 入力と重みの乗算値
        ...,

        # 学習関数 grad_descent_learn(input, goal_pred, pred, weight) の定義
        ...,
        関数名 : grad_descent_learn
        引数 :
            input : 入力値
            goal_pred : 目的値
            pred : 予測値
            weight : 重み
        処理 : 勾配降下法に基づき重みを修正する
        戻り値 : 誤差, 修正された重み
        ...,

        # 重みの初期化

        # 入力値の初期化

        # 目的値の初期化

        # 予測・比較・学習を40回繰り返す

        # 予測を求める
  
```

```
# 学習する
```

```
# 誤差と予測を表示
```

```
[1 回目]: 誤差 = 0.6400000000000001、予測 = 0.2
[2 回目]: 誤差 = 0.5625、予測 = 0.25
[3 回目]: 誤差 = 0.494384765625、予測 = 0.296875
[4 回目]: 誤差 = 0.43451786041259766、予測 = 0.3408203125
[5 回目]: 誤差 = 0.38190046325325966、予測 = 0.38201904296875
[6 回目]: 誤差 = 0.33565470403118525、予測 = 0.4206428527832031
[7 回目]: 誤差 = 0.2950090172149089、予測 = 0.45685267448425293
[8 回目]: 誤差 = 0.25928526903654103、予測 = 0.4907993823289871
[9 回目]: 誤差 = 0.22788744348914738、予測 = 0.5226244209334254
[10 回目]: 誤差 = 0.20029169837913344、予測 = 0.5524603946250863
[11 回目]: 誤差 = 0.17603762552853525、予測 = 0.5804316199610184
[12 回目]: 誤差 = 0.1547205693121892、予測 = 0.6066546437134548
[13 回目]: 誤差 = 0.1359848753720413、予測 = 0.6312387284813639
[14 回目]: 誤差 = 0.11951795686995816、予測 = 0.6542863079512786
[15 回目]: 誤差 = 0.10504507928023663、予測 = 0.6758934137043238
[16 回目]: 誤差 = 0.0923247767111455、予測 = 0.6961500753478035
[17 回目]: 誤差 = 0.08114482328128024、予測 = 0.7151406956385657
[18 回目]: 誤差 = 0.07131869233706269、予測 = 0.7329444021611554
[19 回目]: 誤差 = 0.06268244443687152、予測 = 0.7496353770260832
[20 回目]: 誤差 = 0.055091992180844096、予測 = 0.765283165961953
[21 回目]: 誤差 = 0.048420696252695024、予測 = 0.7799529680893309
[22 回目]: 誤差 = 0.04255725256584524、予測 = 0.7937059075837477
[23 回目]: 誤差 = 0.03740383526294992、予測 = 0.8065992883597635
[24 回目]: 誤差 = 0.03287446458657707、予測 = 0.8186868328372783
[25 回目]: 誤差 = 0.028893572390546244、予測 = 0.8300189057849484
[26 回目]: 誤差 = 0.025394741358878533、予測 = 0.8406427241733891
[27 回目]: 誤差 = 0.02231959689745183、予測 = 0.8506025539125524
[28 回目]: 誤差 = 0.019616833210651007、予測 = 0.8599398942930179
[29 回目]: 誤差 = 0.017241357314048727、予測 = 0.8686936508997043
[30 回目]: 誤差 = 0.015153536701800639、予測 = 0.8769002977184728
[31 回目]: 誤差 = 0.01331853811681697、予測 = 0.8845940291110682
[32 回目]: 誤差 = 0.011705746391733676、予測 = 0.8918069022916264
[33 回目]: 誤差 = 0.010288253664609686、予測 = 0.8985689708983997
[34 回目]: 誤差 = 0.009042410447410849、予測 = 0.9049084102172498
[35 回目]: 誤差 = 0.007947431057294686、予測 = 0.9108516345786717
[36 回目]: 誤差 = 0.006985046827700405、予測 = 0.9164234074175047
[37 回目]: 誤差 = 0.006139201313408554、予測 = 0.9216469444539107
[38 回目]: 誤差 = 0.005395782404362994、予測 = 0.9265440104255412
[39 回目]: 誤差 = 0.004742386878834662、予測 = 0.9311350097739449
[40 回目]: 誤差 = 0.004168113467725774、予測 = 0.9354390716630734
```

問題 2

▶ 問題 1 のプログラムの学習関数 `grad_descent_learn` をホット&コールド学習に従って学習する関数 `hot_and_cold_learn` にコメントに従って変更して実装してください

ただし、重みの微調整量は **0.1** としてください

In [3]:

```
...,
# 空のニューラルネットワーク neural_network の定義
関数名: neural_network
引数: input -> 入力, weight -> 重み
処理: 入力と重みを掛けて返す
戻り値: 入力と重みの乗算値
...,

# 学習関数 hot_and_cold_learn(input, goal_pred, pred, weight) の定義
...,
関数名: hot_and_cold_learn
```

引数：

input：入力値
goal_pred：目的値
pred：予測値
weight：重み

処理：ホット&コールド学習に基づき重みを修正する
戻り値：誤差，修正された重み
...

重みの初期化

入力値の初期化

目的値の初期化

予測・比較・学習を40回繰り返す

```
[1 回目]：誤差 = 0.6400000000000001、予測 = 0.2
[2 回目]：誤差 = 0.6006250000000001、予測 = 0.225
[3 回目]：誤差 = 0.5625、予測 = 0.25
[4 回目]：誤差 = 0.525625、予測 = 0.275
[5 回目]：誤差 = 0.48999999999999994、予測 = 0.30000000000000004
[6 回目]：誤差 = 0.4556249999999999、予測 = 0.32500000000000007
[7 回目]：誤差 = 0.4224999999999999、予測 = 0.35000000000000001
[8 回目]：誤差 = 0.3906249999999999、予測 = 0.37500000000000001
[9 回目]：誤差 = 0.35999999999999998、予測 = 0.400000000000000013
[10 回目]：誤差 = 0.330624999999999984、予測 = 0.425000000000000016
[11 回目]：誤差 = 0.30249999999999998、予測 = 0.45000000000000002
[12 回目]：誤差 = 0.27562499999999998、予測 = 0.47500000000000002
[13 回目]：誤差 = 0.24999999999999978、予測 = 0.50000000000000002
[14 回目]：誤差 = 0.22562499999999977、予測 = 0.52500000000000002
[15 回目]：誤差 = 0.20249999999999976、予測 = 0.55000000000000003
[16 回目]：誤差 = 0.18062499999999976、予測 = 0.57500000000000003
[17 回目]：誤差 = 0.15999999999999975、予測 = 0.60000000000000003
[18 回目]：誤差 = 0.14062499999999975、予測 = 0.62500000000000003
[19 回目]：誤差 = 0.12249999999999975、予測 = 0.65000000000000004
[20 回目]：誤差 = 0.10562499999999976、予測 = 0.67500000000000004
[21 回目]：誤差 = 0.08999999999999976、予測 = 0.70000000000000004
[22 回目]：誤差 = 0.07562499999999976、予測 = 0.72500000000000004
[23 回目]：誤差 = 0.06249999999999978、予測 = 0.75000000000000004
[24 回目]：誤差 = 0.05062499999999979、予測 = 0.77500000000000005
[25 回目]：誤差 = 0.03999999999999981、予測 = 0.80000000000000005
[26 回目]：誤差 = 0.030624999999999823、予測 = 0.82500000000000005
[27 回目]：誤差 = 0.02249999999999984、予測 = 0.85000000000000005
[28 回目]：誤差 = 0.015624999999999861、予測 = 0.87500000000000006
[29 回目]：誤差 = 0.009999999999999884、予測 = 0.90000000000000006
[30 回目]：誤差 = 0.0056249999999999105、予測 = 0.92500000000000006
[31 回目]：誤差 = 0.002499999999999938、予測 = 0.95000000000000006
[32 回目]：誤差 = 0.0006249999999999678、予測 = 0.97500000000000006
[33 回目]：誤差 = 4.437342591868191e-31、予測 = 1.00000000000000007
[34 回目]：誤差 = 0.0006249999999999678、予測 = 0.97500000000000006
[35 回目]：誤差 = 4.437342591868191e-31、予測 = 1.00000000000000007
[36 回目]：誤差 = 0.0006249999999999678、予測 = 0.97500000000000006
[37 回目]：誤差 = 4.437342591868191e-31、予測 = 1.00000000000000007
[38 回目]：誤差 = 0.0006249999999999678、予測 = 0.97500000000000006
[39 回目]：誤差 = 4.437342591868191e-31、予測 = 1.00000000000000007
[40 回目]：誤差 = 0.0006249999999999678、予測 = 0.97500000000000006
```

In []: