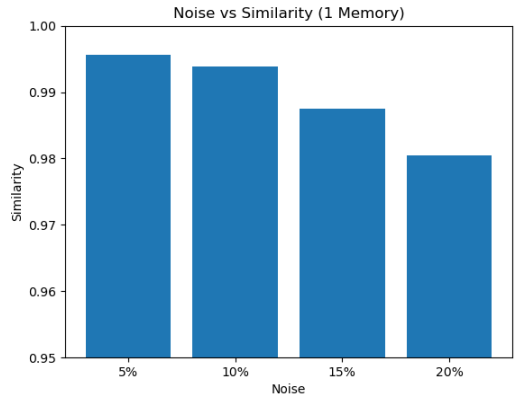
知能機械情報学　レポート課題1

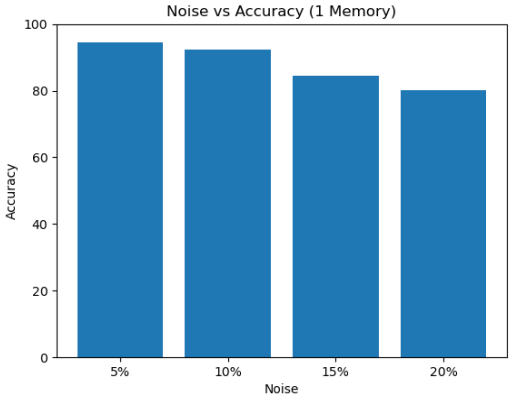
機械情報工学科　03220296 吉川駿

コードURL　：　https://github.com/shuny42657/Neuro_Report_1

第一回授業の説明を参考にし、ホップフィールドネットワークを実装した。一ステップで更新されるノードは、5 × 5 = 25個の中からランダムに毎回選んだ。エネルギー計算等の際に、エネルギーは重みとノードの各成分の積の和のみに基づいて計算し、閾値θは設定しなかった。エネルギーが低い状態に収束したかどうかの判定は、エネルギーの値が150ステップ連続で変化しなかったかどうかで行った（25ノードのうち、1ノードが正解の記憶と異なる値である時、このノードの値が150ステップの間修正されず（更新するランダムなノードとして選ばれない）、エネルギーの値が変化しない確率は、0.96 ** 150 = 0.00219... で、およそ 0.2 %ほどなので、150ステップ連続でエネルギーの値が変化しなかった場合は、ネットワークはすでに定常状態に達していると考えるのは妥当である）。

はじめに、1種類の記憶(アルファベットのSを表した二次元配列、”memory_s”)で、ノイズを5,10,15,20%と、増やしていった際の想起性能を示す(1_memory.py)。以下、本レポートにおいて、想起性能は、類似度（Similarity)と正答率（Accuracy)の二種類で示す。各ノイズにおける試行回数は100回で、類似度は各試行の平均である。

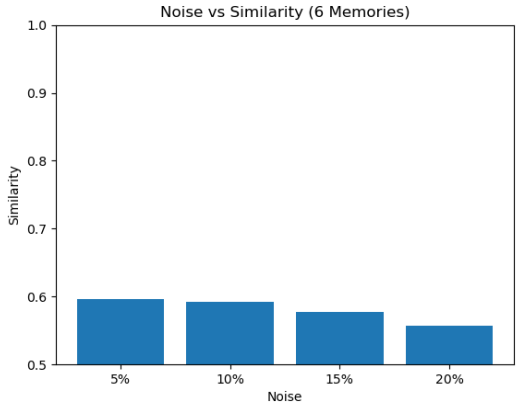





```
Noise  5 %,  SIM : 0.9952000000000001 ACC : 94.0
Noise  10 %,  SIM : 0.9904000000000002 ACC : 88.0
Noise  15 %,  SIM : 0.9896000000000003 ACC : 88.0
Noise  20 %,  SIM : 0.9856000000000003 ACC : 82.0
```

注）（SIM, ACCはそれぞれ、類似度と正答率を表す）

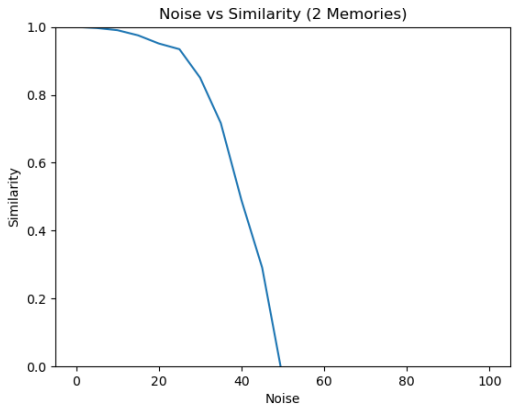
次に、記憶を六種類に増やして、同じ実験を行う(6_memories.py)。ここで用意した六種類の記憶は、アルファベットのS,H,U,N そして、数字の1、2を表した二次元配列である（変数名はコード参照）。”memory_s” のみにノイズを加えて、想起性能を検証した結果が以下である。

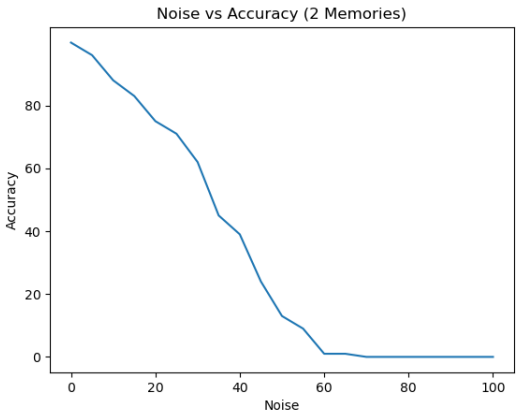



```
Noise  5 %,  SIM : 0.5968000000000009 ACC : 0.0
Noise  10 %,  SIM : 0.594400000000001 ACC : 0.0
Noise  15 %,  SIM : 0.584800000000001 ACC : 0.0
Noise  20 %,  SIM : 0.5804000000000009 ACC : 0.0
```

本実験では、類似度の値が、記憶が一種類の時に比べて半程度に低下し、正答率に関しては、いずれのノイズに設定した場合でも0%になった（正答率のグラフは割愛）。

続いて、記憶が2種類、4種類のそれぞれの場合において、ノイズを0%から5%刻みで100%まで増加させた時の、想起性能の変化を検証した。
まず、記憶が2種類だった場合の結果を以下に示す(2_memories.py)。

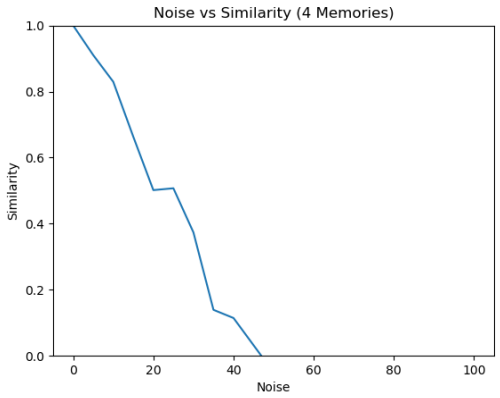


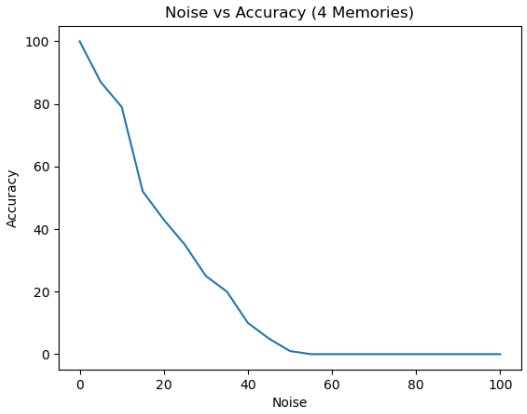


```
Noise  0 %,  SIM : 1.0 ACC : 100.0
Noise  5 %,  SIM : 0.9968 ACC : 96.0
Noise  10 %,  SIM : 0.9904000000000002 ACC : 88.0
Noise  15 %,  SIM : 0.9752000000000003 ACC : 83.0
Noise  20 %,  SIM : 0.9512000000000004 ACC : 75.0
Noise  25 %,  SIM : 0.9344000000000005 ACC : 71.0
Noise  30 %,  SIM : 0.8504000000000002 ACC : 62.0
Noise  35 %,  SIM : 0.7168000000000002 ACC : 45.0
Noise  40 %,  SIM : 0.4896000000000002 ACC : 39.0
Noise  45 %,  SIM : 0.29120000000000007 ACC : 24.0
Noise  50 %,  SIM : -0.0352 ACC : 13.0
Noise  55 %,  SIM : -0.2920000000000001 ACC : 9.0
Noise  60 %,  SIM : -0.48560000000000014 ACC : 1.0
Noise  65 %,  SIM : -0.7632000000000003 ACC : 1.0
Noise  70 %,  SIM : -0.8208000000000003 ACC : 0.0
Noise  75 %,  SIM : -0.9168000000000004 ACC : 0.0
Noise  80 %,  SIM : -0.9800000000000003 ACC : 0.0
Noise  85 %,  SIM : -0.9688000000000002 ACC : 0.0
Noise  90 %,  SIM : -0.9952000000000001 ACC : 0.0
Noise  95 %,  SIM : -0.9976 ACC : 0.0
Noise  100 %,  SIM : -1.0 ACC : 0.0
```

類似度は、ノイズが30%のあたりで急激に下がり始め、50%のところで限りなく0に近づく。その後、類似度は負の値をとり、100%の時の類似度はちょうど−1になる。ノイズn = 50%以降は、各ノードの値を反転させれば、ノイズ （100 - n)%と同じ問題設定になる。よって、ノイズ100%と0%の時では、結果が反転しているだけで、想起性能としては実質的に同等だと考えることができる。

次に、記憶の種類を4種類に増やした際の結果を示す(4_memories.py)。






```
Noise  0 %,  SIM : 1.0 ACC : 100.0
Noise  5 %,  SIM : 0.9104000000000002 ACC : 87.0
Noise  10 %,  SIM : 0.8296000000000002 ACC : 75.0
Noise  15 %,  SIM : 0.6631999999999999 ACC : 56.0
Noise  20 %,  SIM : 0.5016 ACC : 36.0
Noise  25 %,  SIM : 0.5072000000000001 ACC : 37.0
Noise  30 %,  SIM : 0.3736000000000004 ACC : 22.0
Noise  35 %,  SIM : 0.13919999999999993 ACC : 11.0
Noise  40 %,  SIM : 0.11439999999999992 ACC : 6.0
Noise  45 %,  SIM : 0.032 ACC : 1.0
Noise  50 %,  SIM : -0.0504 ACC : 0.0
Noise  55 %,  SIM : -0.05200000000000005 ACC : 0.0
Noise  60 %,  SIM : -0.14400000000000002 ACC : 0.0
Noise  65 %,  SIM : -0.2135999999999999 ACC : 0.0
Noise  70 %,  SIM : -0.34239999999999987 ACC : 0.0
Noise  75 %,  SIM : -0.384 ACC : 0.0
Noise  80 %,  SIM : -0.5463999999999999 ACC : 0.0
Noise  85 %,  SIM : -0.7960000000000003 ACC : 0.0
Noise  90 %,  SIM : -0.7960000000000003 ACC : 0.0
Noise  95 %,  SIM : -0.9648 ACC : 0.0
Noise  100 %,  SIM : -1.0 ACC : 0.0
```

記憶の種類が増えた方が、想起性能の低下率が大きくなった（特に類似度において顕著である）。

（考察、講義の感想）
人間の脳の挙動をモデル化して機械学習の１手法として利用するという考え方は大変興味深かった。ただ、記憶の種類が少ない時は想起性能も十分にあるものの、記憶の種類がそこから少し増えただけで、かなり激しく性能が低下してしまった。25マスで表現できる記憶には限度があり、想起性能はモデルが表現できる記憶の数に依存すると考えられるので、仮に、ノードの数を増やすなどすれば想起性能の向上が期待できるだろう。
ホップフィールドネットワークの他にも授業で紹介された学習手法を自分で実装してみて、性能や結果を検証してみたいと思った。