

## CP II – Projekt 4: Neuronale Netze

– Benjamin Dummer (532716) –

Datum: 31. März 2018

### Aufgabe 4.1: Hopfield-Modell mit asynchronem und synchronem Update

#### Aufgabenstellung:

Das Hopfield-Modell soll, wie in der Literatur ([1], [2]) beschrieben, mit asynchronem und synchronem Update realisiert werden. Das Netzwerk soll für verschiedene Anzahlen gespeicherter Bilder untersucht werden.

#### Lösung:

Die Umsetzung ist in dem Python-Skripten `task_4-1.py` & `task_4-1_big-run.py` und der dazugehörigen Funktionenbibliothek `hopfield_func.py` zu finden. Als Standard-Parameter werden in der gesamten Bearbeitung folgende Werte verwendet:

- Anzahl Neuronen:  $N = 100$
- Anzahl gespeicherter Bilder:  $P \in \{10, 20, 30\}$

Anhand verschiedener Testläufe für die Konvergenz mit synchronem und asynchronem Update wird die maximale Iterationszahl auf 100 festgelegt, da eine Konvergenz, wenn sie eintritt, deutlich schneller erreicht ist.

Die folgende Tabelle zeigt die gemittelten Ergebnisse für die relativen Fehler (Hamming-Abstände) nach einer Iteration  $\delta_1$  und nach der Konvergenz  $\delta_{\text{inf}}$  sowie die Anzahl Iterationen bis zur Konvergenz  $N_{\text{inf}}$  für die verschiedenen Anzahlen gespeicherter Bilder bei asynchronem Update:

$P$	$\delta_1[\%]$	$\delta_{\text{inf}}[\%]$	$N_{\text{inf}}$
10	0.07	0.2	1.1
20	1.8	7	3.0
30	5.4	21	5.8

Die einzelnen Werte sind ohne Fehlerangaben, da es keine explizite Fehleranalyse gab. Hier wurden jeweils 100 verschiedene Bilderkonfigurationen gemittelt. Durch verschiedene Testläufe zeigte sich, dass insbesondere für  $P = 10$  der relative Fehler der einzelnen Werte durchaus bei 50% liegen kann. Trotzdem lässt sich eindeutig ein Trend erkennen. Schaut man sich zusätzlich nicht gemittelte Datensätze genauer an, ist eindeutig zu erkennen, dass für  $P = 10$  nur sehr selten eine Veränderung des Netzwerks auftritt. Eine Iteration bedeutet im Zusammenhang mit dem Abbruchkriterium, dass das Netzwerk vor und nach der Iteration exakt den gleichen Zustand hat.

Für den Fall des synchronen Updates fällt auf, dass die Fehler und die Konvergenzgeschwindigkeit (wenn eine Konvergenz eintritt) grundsätzlich ähnlich bleiben. Jedoch tritt nicht immer eine Konvergenz ein und der Anteil der nicht konvergierenden Versuche steigt mit der Anzahl der gespeicherten Bilder.

## Aufgabe 4.2: Konvergenz und endliche Temperatur

### Aufgabenstellung:

Für das Hopfield-Modell mit asynchronem Update soll das Konvergenzverhalten zu gespeicherten Bildern untersucht werden und eine endliche Temperatur implementiert werden.

### Lösung:

Die Umsetzung ist in dem Python-Skripten `task_4-2.py` & `task_4-2_big-run.py` und der dazugehörigen Funktionenbibliothek `hopfield_func.py` zu finden. Es wurden die in Aufgabe 4.1 benannten Parameter weiter genutzt. Zusätzlich wurde die Fehlertoleranz für eine Konvergenz auf 5% festgelegt.

Bei Testläufen mit der vorgeschlagenen Anzahl von zufälligen Start-Konfigurationen wurde festgestellt, dass die Ergebnisse sehr stark streuen und es somit eine starke Abhängigkeit des Konvergenzverhaltens von dem Set gespeicherter Bilder vs. Startkonfiguration gibt. Außerdem war schnell zu erkennen, dass für  $P \in \{20, 30\}$  überwiegend spurious states erreicht werden und es nur sehr wenig „Wiedererkennung“ im Netzwerk gibt. Dies deckt sich mit der Erwartung, dass ein Hopfield-Netzwerks nur mit  $P < 0.138N$  einen nützlichen Speicher darstellt (Stichwort: Kapazität, [2]). Um genauere Aussagen über den Anteil von spurious states zu erhalten, muss die Statistik für die Simulation erhöht werden. Genauso kann auf den zweiten Teil der Frage, ob bei der Implementation von endlicher Temperatur ein  $\beta$  existiert, für dass sich der Anteil der spurious states verringert, nur mit größerer Statistik eingegangen werden. Für die Plots in Abb. 1 wurden jeweils 1000 zufällige Startkonfigurationen für 1000 verschiedene Sets gespeicherter Bilder simuliert. Dabei wurden unterschiedliche „Dauern“ der endlichen Temperatur untersucht. Der Anteil der spurious states für das Netzwerk ohne endliche Temperatur ist jeweils durch die gestrichelte, schwarze Linie dargestellt. Auch für diese Ergebnisse wurde keine aufwändige Fehleruntersuchung durchgeführt, jedoch erlaubt der Verlauf der Kurven einen qualitativen Eindruck über die Schwankungen der gemessenen Werte.

Zunächst muss festgestellt werden, dass in Abb. 1 *Oben* die Kurve für 100 Iterationen mit endlicher Temperatur fehlerhaft ist, da vergessen wurde, das Konvergenzlimit zu verändern. Trotzdem liegen die Werte für  $\beta > 4$  vermutlich nah an den tatsächlichen, da sich die Zustände bei ausreichend kleinem Rauschen nach so vielen Iterationen sehr nah am Minimum im Zustandsraum befinden. Leider blieb keine Zeit, um noch eine Simulation für eine Korrektur der Werte anzustoßen.

Insgesamt ist für Abb. 1 *Oben* und *Mitte* (für letztere nur in gewissem Rahmen) zu erkennen, dass die Dauer der endlichen Temperatur einen deutlichen Einfluss auf die Ergebnisse hat. So tritt das Minimum in der Kurve erst für eine ausreichend lange Dauer sichtbar auf. Es zeigt sich auch, dass ein starkes Rauschen (kleines  $\beta$ ) zu einer Erhöhung des Anteils der spurious states führt (im Vergleich zum System ohne endliche Temperatur). Außerdem nähern sich die Kurven, wie zu erwarten, für  $\beta \rightarrow \infty$  dem Wert für das System ohne endliche Temperatur an. Für Abb. 1 *Unten* hat keiner der Parameter einen Einfluss auf das Ergebnis.

Für  $P = 10$  existiert also ein  $\beta_{min}$  für das der Anteil der spurious states gesenkt wird mit ungefähr  $4 < \beta_{min} < 9$ . Für diese Anzahl gespeicherter Bilder ist ein Optimum von 30 – 40% für den Anteil von spurious states zu erwarten.

Auch für  $P = 20$  existiert ein solches  $\beta_{min}$  mit ungefähr  $5 < \beta_{min} < 18$ . Dazu sollte allerdings festgestellt werden, dass die Daten für  $P = 20$  eine recht geringe Aussagekraft haben, da es sich um minimale Änderungen handelt.

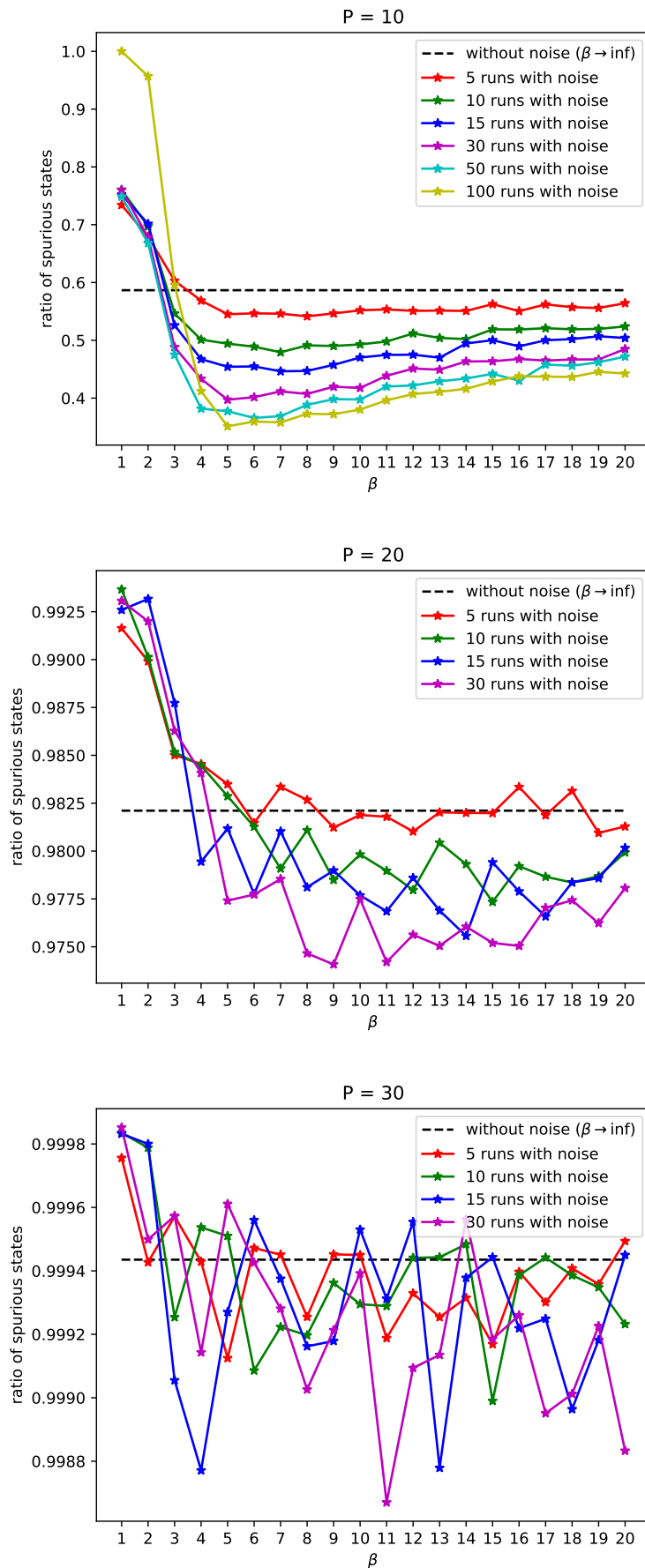


Abbildung 1: Zusammenhang zwischen der inversen Temperatur  $\beta$  und dem Anteil der „spurious states“ für verschiedene Anzahlen von Iterationen mit endlicher Temperatur zu Beginn; *Oben:*  $P = 10$ , *Mitte:*  $P = 20$ , *Unten:*  $P = 30$

## Literatur

- [1] U. Wolf und B. Bunk, „CP II - Skript“, Humboldt-Universität zu Berlin, 2003
- [2] J. Hertz, A. Krogh und R. Palmer, „Introduction to the Theory of Neural Computation“, Addison-Wesley, 1991