

Optional: differenzierte Initialisierung der Gewichte

Dieser Teil ist optional, da es hier lediglich um eine einfache, wenn auch beliebte, Verfeinerung der Gewichtsinitialisierung geht.

Wenn Sie die Diskussion am Ende von Kapitel 1 dieses Buchs aufmerksam verfolgt haben, bei der es um die Vorbereitung der Daten und die Initialisierung der Gewichte ging, wissen Sie, dass es auch Verfechter eines etwas komplexeren Ansatzes gibt, um die zufälligen Anfangsgewichte zu erzeugen. Sie entnehmen Stichproben der Gewichte aus einer Normalverteilung mit dem Mittelwert null und einer Standardabweichung, die sich auf die Anzahl der Verknüpfungen zu einem Knoten bezieht: $1/\sqrt{\text{Anzahl der eingehenden Verknüpfungen}}$.

Mit der `numpy`-Bibliothek ist das leicht zu realisieren. Auch hier ist Google hilfreich, um die richtige Dokumentation zu finden. Die Funktion `numpy.random.normal()`, die Stichproben aus einer Normalverteilung entnimmt, wird unter <http://docs.scipy.org/doc/numpy-1.10.1/reference/generated/numpy.random.normal.html> beschrieben. Als Parameter übernimmt die Funktion den Mittelwert der Verteilung, die Standardabweichung und die Größe eines `numpy`-Arrays, falls die Funktion eine Matrix mit Zufallszahlen und nicht nur eine einzelne Zahl erzeugen soll.

Der aktualisierte Code für die Initialisierung der Gewichte sieht folgendermaßen aus:

```
self.wih = numpy.random.normal(0.0, pow(self.hnodes, -0.5), (self.hnodes, self.inodes))
self.who = numpy.random.normal(0.0, pow(self.onodes, -0.5), (self.onodes, self.hnodes))
```

Den Mittelwert der Normalverteilung haben wir auf 0,0 gesetzt. Der Ausdruck für die Standardabweichung – bezogen auf die Anzahl der Knoten in der nächsten Schicht – hat in Python die Form `pow(self.hnodes, -0.5)`. Es wird also die Anzahl der Knoten hoch $-0,5$ genommen, d. h. die Wurzel aus der Anzahl der Knoten gezogen. Der letzte Parameter gibt die gewünschte Gestalt des `numpy`-Arrays an.