Unterprogramme und Definitionen zum MATHCAD-Dokument *Neuronales_Netz_Buchstaben.mcd*
Muss als Verweis (Einfügen -> Verweis) in das Hauptdokument eingefügt werden (siehe dort!)!

Dieses Mathcad-Dokument enthält diverse Definitionen, Hilfsfunktionen, Unterprogramme sowie den eigentlichen Backpropagation-Algorithmus.

Matrixdefinition der Zeichen:
Ihre grafische Darstellung ist gesperrt, damit sie nicht unabsichtlich verändert wird. Die Bedeutung der Zahlen 0 und 1 ist bei genauerem Hinsehen offensichtlich.

$$Z_0 := \begin{pmatrix} 0&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1 \end{pmatrix} \quad Z_1 := \begin{pmatrix} 1&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&0 \end{pmatrix} \quad Z_2 := \begin{pmatrix} 0&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&1\\ 0&1&1&1&0 \end{pmatrix} \quad Z_3 := \begin{pmatrix} 1&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&0 \end{pmatrix} \quad Z_4 := \begin{pmatrix} 1&1&1&1&1\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&1&1&1&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&1&1&1&1 \end{pmatrix}$$

$$Z_5 := \begin{pmatrix} 1&1&1&1&1\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&1&1&1&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0 \end{pmatrix} \quad Z_6 := \begin{pmatrix} 0&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&0\\ 1&0&1&1&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 0&1&1&1&0 \end{pmatrix} \quad Z_7 := \begin{pmatrix} 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1 \end{pmatrix} \quad Z_8 := \begin{pmatrix} 0&1&1&1&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&1&1&1&0 \end{pmatrix} \quad Z_9 := \begin{pmatrix} 0&0&1&1&1\\ 0&0&0&1&0\\ 0&0&0&1&0\\ 0&0&0&1&0\\ 0&0&0&1&0\\ 1&0&0&1&0\\ 0&1&1&0&0 \end{pmatrix}$$

$$Z_{10} := \begin{pmatrix} 1&0&0&0&1\\ 1&0&0&1&0\\ 1&0&1&0&0\\ 1&1&0&0&0\\ 1&0&1&0&0\\ 1&0&0&1&0\\ 1&0&0&0&1 \end{pmatrix} \quad Z_{11} := \begin{pmatrix} 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&1&1&1&1 \end{pmatrix} \quad Z_{12} := \begin{pmatrix} 1&0&0&0&1\\ 1&1&0&1&1\\ 1&0&1&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1 \end{pmatrix} \quad Z_{13} := \begin{pmatrix} 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&0&0&1\\ 1&0&1&0&1\\ 1&0&0&1&1\\ 1&0&0&0&1\\ 1&0&0&0&1 \end{pmatrix} \quad Z_{14} := \begin{pmatrix} 0&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 0&1&1&1&0 \end{pmatrix}$$

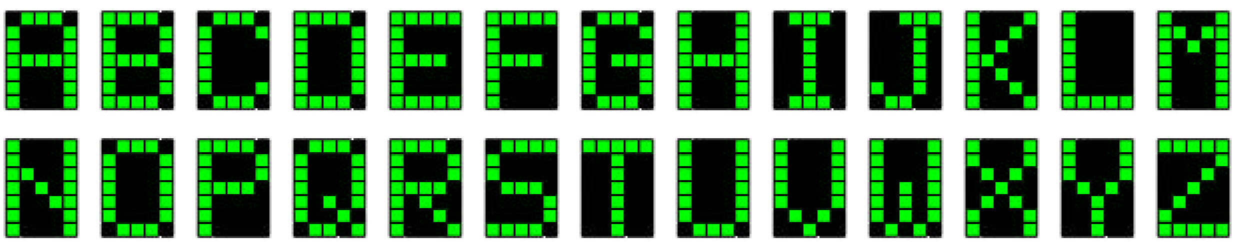
$$Z_{15} := \begin{pmatrix} 1&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0 \end{pmatrix} \quad Z_{16} := \begin{pmatrix} 0&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&0&1&0&1\\ 1&0&0&1&0\\ 0&1&1&0&1 \end{pmatrix} \quad Z_{17} := \begin{pmatrix} 1&1&1&1&0\\ 1&0&0&0&1\\ 1&0&0&0&1\\ 1&1&1&1&0\\ 1&0&1&0&0\\ 1&0&0&1&0\\ 1&0&0&0&1 \end{pmatrix} \quad Z_{18} := \begin{pmatrix} 0&1&1&1&1\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 0&1&1&1&0\\ 0&0&0&0&1\\ 0&0&0&0&1\\ 1&1&1&1&0 \end{pmatrix} \quad Z_{19} := \begin{pmatrix} 1&1&1&1&1\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0\\ 0&0&1&0&0 \end{pmatrix}$$

$$Z_{20} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad Z_{21} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad Z_{22} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad Z_{23} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad Z_{24} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$Z_{25} := \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$Zeichenzahl := länge(Z)$

$Zeichenzahl = 26$



$f\_rand\_wv(x, y) := rnd(2) - 1$

Für die Initialisierung der Gewichtsmatrizen mit Zufallszahlen zwischen -1 und 1.

$f\_null(x, y) := 0$

$$mat\_to\_vek(mat) := \left| \begin{array}{l} a \leftarrow mat^{\langle 0 \rangle} \\ \text{for } j \in 1 .. spalten(mat) - 1 \\ \quad a \leftarrow stapeln\left(a, mat^{\langle j \rangle}\right) \\ a \end{array} \right.$$

Wandelt die Matrixdarstellung des Zeichens in die Vektordarstellung um durch Übereinanderstapeln der einzelnen Matrixspalten.

$$make\_train(Zeichenzahl) := \left| \begin{array}{l} train \leftarrow mat\_to\_vek\left(Z_0\right) \\ \text{for } j \in 1 .. Zeichenzahl - 1 \\ \quad train \leftarrow erweitern\left(train, mat\_to\_vek\left(Z_j\right)\right) \\ train \end{array} \right.$$

Erzeugt die Trainingsmatrix. Ihre Spalten enthalten die einzelnen Trainingsmuster.

$$zeichen\_auswertung(v) := \left| \begin{array}{l} mx \leftarrow max(v) \\ for\ \ i \in 0..\,zeilen(v) - 1 \\ \quad break\ \ if\ \ mx = v_i \\ i \end{array} \right.$$

Berechnet den Index jenes Zeichens, das zum größten Wert eines Ausgangsneurons gehört.

$$akt(x) := \frac{1}{1 + exp(-x)}$$

Aktivierungsfunktion

Das Hauptprogramm: der Backpropagation-Algorithmus.
Es wurde intensiver Gebrauch gemacht von in Mathcad eingebauten Matrixfunktionen, was die Formulierung recht kompakt, aber eher schwer verständlich macht; die Ausführungsgeschwindigkeit konnte dadurch (wie sich im Lauf der Programmentwicklung gezeigt hat) jedoch drastisch gesteigert werden.

$$back\_prop(w, v, train, y\_s, \alpha, N, akt, \varepsilon) := \left| \begin{array}{l} z1 \leftarrow w^T \cdot train \\ z \leftarrow \overrightarrow{akt(z1)} \\ y1 \leftarrow v^T \cdot z \\ y \leftarrow \overrightarrow{akt(y1)} \\ diff \leftarrow y\_s - y \\ diff2 \leftarrow \overrightarrow{(diff \cdot diff)} \\ F\_start \leftarrow \frac{1}{2} \cdot \sum_{k=0}^{Zeichenzahl-1} \left(\left| diff^{\langle k \rangle} \right| \right)^2 \\ for\ \ n \in 0..\,N - 1 \\ \quad \left| \begin{array}{l} z1 \leftarrow w^T \cdot train \\ z \leftarrow \overrightarrow{akt(z1)} \\ y1 \leftarrow v^T \cdot z \\ y \leftarrow \overrightarrow{akt(y1)} \\ diff \leftarrow y\_s - y \\ diff2 \leftarrow \overrightarrow{(diff \cdot diff)} \\ F\_akt \leftarrow \frac{1}{2} \cdot \sum_{k=0}^{Zeichenzahl-1} \left(\left| diff^{\langle k \rangle} \right| \right)^2 \\ break\ \ if\ \ F\_akt < \varepsilon \\ t \leftarrow \overrightarrow{[\,diff \cdot [\,[\,y \cdot (1 - y)]\,]\,]} \\ \Delta v \leftarrow z \cdot t^T \\ v \leftarrow v + \Delta v \cdot \alpha \\ temp \leftarrow v \cdot t \end{array} \right. \end{array} \right.$$

$$\Delta w \leftarrow train \cdot \left(\overrightarrow{[z \cdot (1 - z) \cdot temp]}\right)^T$$
$$w \leftarrow w + \Delta w \cdot \alpha$$
$$wv \leftarrow stapeln\left(w, v^T\right)$$
$$zei \leftarrow zeilen(wv)$$
$$wv_{zei,0} \leftarrow n$$
$$wv_{zei,1} \leftarrow F\_start$$
$$wv_{zei,2} \leftarrow F\_akt$$
$$wv$$

Die Prozedur berechnet zuerst den Anfangsfehler und führt danach die Iterationen aus, wobei jedesmal der aktuelle Fehler berechnet wird. Bei Erreichen der maximalen Anzahl der Lernzyklen (N) oder wenn der Fehler die vorgegebene Fehlerschranke unterschreitet, wird die Berechnung abgebrochen.
Die beiden Gewichtsmatrizen, die Zahl der Iterationen, Anfangs- und Endfehler werden in eine einzige Matrix gepackt und zurückgegeben.