

## Erläuterungen zu den Klassenmodulen *clsMatrix*, *clsVektor* und *clsZelle*

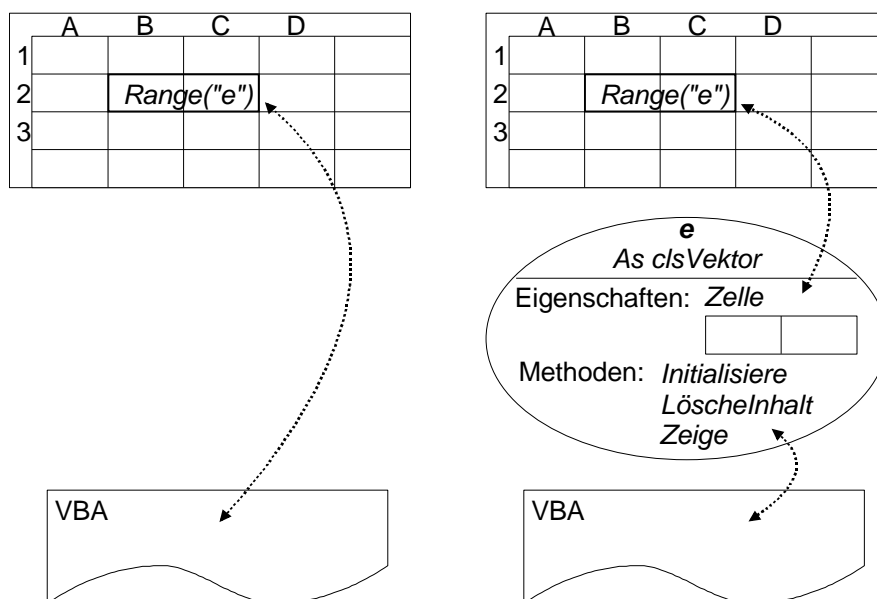
Das Lernen eines zweistufigen neuronalen Netzes mit dem Fehlerrückführungs-Verfahren ist rechenintensiv. Im Vergleich zum Perzeptron ist die Zahl der Lernschritte meist wesentlich höher. Während wir bei der Implementation des Perzeptrons die einzelnen Schritte noch direkt in den Zellen eines Excel-Tabellenblattes berechnen konnten, führt derselbe Ansatz bei der Fehlerrückführung zu langen Antwortzeiten.

Der Lernprozess lässt sich unter anderem durch eine effizientere Berechnung der Lernschritte beschleunigen. Langsam sind vor allem Zugriffe auf die Zellen des Tabellenblatts. Wir versuchen deshalb möglichst viele Zwischenergebnisse in Programmvariablen anstatt in Tabellenblattzellen zu speichern.

Die Klassen *clsMatrix*, *clsVektor* und *clsZellen* verwalten Kopien ausgewählter Zellbereiche in Programmvariablen. Sie unterscheiden sich voneinander nur in Bereichsdimensionen:

- *clsMatrix* operiert auf einem zweidimensionalen Zellbereich (z.B. einer Gewichtematrix). Eine einzelne Zelle wird durch zwei Indizes (bei 1 beginnend) identifiziert.
- *clsVektor* verwaltet einen eindimensionalen horizontalen oder vertikalen Zellbereich (z.B. ein Eingabevektor). Für den Zugriff auf eine Zelle genügt die Angabe eines einzigen Index (bei 1 beginnend).
- *clsZelle* verwaltet den Inhalt einer einzelnen Zelle (z.B. die Lernkonstante). Ihr Wert kann ohne die Angabe eines Index gelesen oder geändert werden.

Wir erläutern die Verwendung der drei Klassen an einem Eingabevektors *e* mit zwei Komponenten, einer 2×2 Gewichtematrix *G* und einer Konstanten *alpha*. Die nachfolgende Abbildung und die anschliessenden Abschnitte stellen die bisherige und die neue Arbeitsweise gegenüber.



### Variablendeklaration und -initialisierung

Um einfach und übersichtlich auf ausgewählte Bereiche des Tabellenblatts „Benutzeroberfläche“ zugreifen zu können, hatten wir beim Perzeptron Modulvariablen vom Typ `Range` deklariert und diesen bei der Initialisierung den entsprechenden Bereich zugewiesen.

Diese Variablen sind nun vom Typ `clsMatrix`, `clsVektor` oder `clsZelle` statt vom Typ `Range`. Zum Beispiel:

<i>Perzeptron:</i>	<b>Private</b> alpha <b>As</b> Range	<i>'Lernrate</i>
	<b>Private</b> e <b>As</b> Range	<i>'Eingabevektor</i>
	<b>Private</b> G <b>As</b> Range	<i>'Gewichtematrix</i>
 <i>Backpropagation:</i>	<b>Private</b> alpha <b>As New</b> clsZelle	<i>'Lernrate</i>
	<b>Private</b> e <b>As New</b> clsVektor	<i>'Eingabevektor</i>
	<b>Private</b> G <b>As New</b> clsMatrix	<i>'Gewichtematrix</i>

Die Zuweisung der ausgewählten Zellbereiche erfolgt wie beim einstufigen Perzeptron in der Prozedur Initialisieren(). Neu muss angegeben werden, ob der aktuelle Inhalt der Zellen übernommen oder gelöscht werden soll. In der Implementation definieren wir hierfür zwei Konstanten löscheWerte bzw. behalteWerte. Zum Beispiel:

<i>Perzeptron:</i>	<b>With</b> Worksheets("Benutzeroberfläche")	
	<b>Set</b> alpha = .Range("alpha")	<i>'Lernrate</i>
	<b>Set</b> e = .Range("e")	<i>'Eingabevektor</i>
	<b>Set</b> G = .Range("G")	<i>'Gewichtematrix</i>
	...	
	<b>End With</b>	
 <i>Backpropagation:</i>	<b>With</b> Worksheets("Benutzeroberfläche")	
	alpha.Initialisiere .Range("alpha"), behalteWerte	<i>'Lernrate</i>
	e.Initialisiere .Range("e"), behalteWerte	<i>'Eingabevektor</i>
	G.Initialisiere .Range("G"), behalteWerte	<i>'Gewichtematrix</i>
	...	
	<b>End With</b>	

## Zugriff und Anzeigen des Inhalts

Auf die Zellen kann nun in gewohnter Weise zugegriffen werden. Anstelle der Methode Cells() der Range-Objekte tritt die Methode Zelle() der Klassen clsZelle, clsVektor und clsMatrix. Zu beachten ist, dass im Fall des Perzeptrons jede Änderung des Inhalts einer Tabellenblattzelle automatisch angezeigt wird. Im Fall der Backpropagation muss eine Aktualisierung dagegen explizit durch Anwenden der Methode Zeige() ausgelöst werden. Z.B.:

<i>Perzeptron:</i>	G.Cells(i, j) = G.Cells(i, j) + alpha * e.Cells(i) * (k.Cells(j) – a.Cells(j))	
	<i>'--- neue Gewichtematrix wird automatisch angezeigt</i>	
 <i>Backpropagation:</i>	G.Zelle(i, j) = G.Zelle(i, j) + alpha.Zelle * e.Zelle(i) * (k.Zelle(j) – a.Zelle(j))	
	G.Zeige	<i>'--- neue Gewichtematrix anzeigen</i>

## Andere Operationen

Für die Implementation des neuronalen Netzes mit Fehlerrückführung benötigen wir eine Möglichkeit, um einen Zellbereich zu löschen. Range-Objekte stellen dafür die Methode ClearContents() bereit. Die entsprechenden Methoden der Klassen clsZelle, clsVektor und clsMatrix heißen LöscheInhalt(). Z.B.:

<i>Perzeptron:</i>	e.ClearContents	<i>'--- Lösche alle Zellen des Eingabevektors</i>
 <i>Backpropagation:</i>	e.LöscheInhalt	<i>'--- Lösche alle Zellen des Eingabevektors</i>

Beim Perzeptron wurde von der Möglichkeit Gebrauch gemacht, Tabellenblattzellen zu formatieren. Diese Möglichkeit bieten die drei Klassen nicht. Z.B.:

<i>Perzeptron:</i>	e.Cells(i).Font.Bold = True	<i>'--- stelle i-te Komponente des Vektors e fett dar</i>
 <i>Backpropagation:</i>	<i>'--- es existiert keine Methode zur Änderung der Darstellung von Zellen</i>	