

Einleitung

Principal Components Analysis

In diesem Abschnitt werden wir uns mit einer spezielleren Variante neuronaler Netze, den PCA-Netzen (**P**incipal **C**omponents **A**alysis), beschäftigen. Diese Netze haben die Eigenschaft, daß Sie bei Eingabe mehrerer Datenvektoren selbständig eine Dekorrelation der Vektorkomponenten lernen können. Es werden also *lineare Abhängigkeiten* zwischen den einzelnen Vektorkomponente beseitigt. Insbesondere handelt es sich dabei um ein *unüberwachtes Lernverfahren*.

Die Dekorrelation wird dadurch erreicht, daß das Netz eine lineare Transformation lernt, durch die die Menge der Eingabevektoren (die man sich häufig einfach als Punktwolke vorstellen kann) möglichst gut in die Richtung der Achsen des Koordinatensystems rotiert wird. Siehe dazu Abbildung 1.

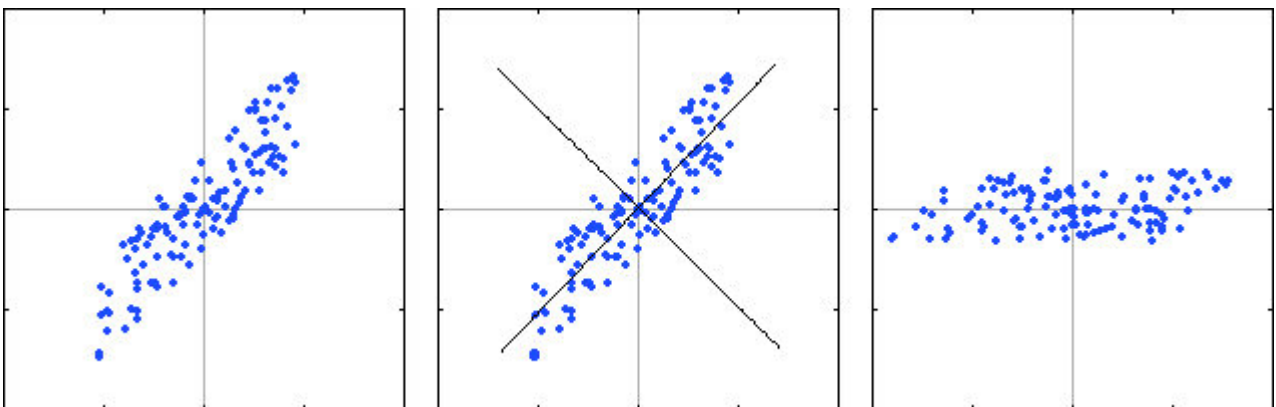


Abbildung 1. Links: Datenpunkte mit korrelierten xy-Koordinaten. Mitte: Die lineare Transformation, die von einem PCA-Netz anhand

der Daten gelernt wurde. Dargestellt sind die neuen Hauptachsen, die nach Anwenden der Transformation angenommen werden und die Punktmenge dekorrelieren. Bei der linearen Transformation handelt es sich stets um eine geeignete Rotation um den Nullpunkt. *Rechts:* Punktmenge nach Durchlaufen eines PCA-Netzes – die linearen Abhängigkeiten zwischen den x- und y-Koordinaten wurden ohne Informationsverlust beseitigt. Die meiste Strukturinformation über das Aussehen der Punktwolke ist nun in der x-Komponente der Punkte konzentriert.

Die Fähigkeit, Daten zu dekorrelieren, ist äußerst nützlich und besitzt eine Reihe von interessanten Anwendungsmöglichkeiten, die wir uns später noch genauer ansehen werden. Allen voran läßt sie Rückschlüsse darauf zu, welche Vektorkomponenten sich durch eine Linearkombination anderer Komponenten ausdrücken lassen und damit keine zusätzliche Information über diesen Datenvektor tragen. Außerdem lassen sich mit bestimmten PCA-Netzen nach der Transformation Komponenten bestimmen, die nur wenig zur Beschaffenheit der Datenvektoren beitragen. In beiden Fällen können diese Komponenten dann vernachlässigt werden, so daß man es bei einer weiteren Analyse nur noch mit niedrigdimensionaleren Vektoren zu tun hat. Dieses Vorgehen bezeichnet man daher auch als *Dimensionsreduktion*. Insbesondere kann man es sehr effektiv als Vorstufe eines (verlustbehafteten) Datenkompressionssystems einsetzen (für natürliche Bilder, Audiodaten, Sensorwerte,...).

Kommen wir nun zur Funktionsweise eines PCA-Netzes. Es gibt mehrere Typen von Netzen, mit denen sich eine PCA durchführen läßt, das interessanteste darunter dürfte jedoch das sogenannte *Sangernetz* sein, wie es 1988 von Terence D. Sanger vom MIT gefunden und in einem Paper von 1989 veröffentlicht wurde. Wie wir noch sehen werden, besitzt es einige recht attraktive Eigenschaften, die es sehr nützlich für die oben beschriebene Analyse von Daten macht.

nach oben

Lernen des ersten Gewichtsvektors

Wir haben uns zuvor mit der Hebb'schen Lernregel beschäftigt, die für ein

einzelnes Neuron mit Gewichtsvektor $\underline{w} \in \mathbb{R}^n$, der Eingabe $\underline{x} \in \mathbb{R}^n$ und der Ausgabe $y = \underline{x}^T \underline{w}$ (wir betrachten hier ein Neuron mit linearer Ausgabefunktion), sowie der Lernrate γ die Form hat:

$$\underline{w} \leftarrow \underline{w} + \gamma \cdot \underline{x} \cdot y$$

Ein Problem dieser Lernregel ist jedoch, daß für konstantes γ die Gewichte unbeschränkt wachsen können, was biologisch nicht sehr plausibel ist. Um dies zu vermeiden, können wir nun fordern, daß sich die Länge von \underline{w} nach einem Lernschritt nicht ändern soll und \underline{w} dazu nach Anwenden der Lernregel normieren. Dies führt zu:

$$\begin{aligned} \underline{w} &\leftarrow \underline{w} + \gamma \cdot \underline{x} \cdot y \\ \underline{w} &\leftarrow \underline{w} \cdot \frac{1}{\|\underline{w}\|_2} \end{aligned}$$

Nehmen wir nun an, wir haben eine Reihe von Eingabevektoren

$\underline{x}^{(1)}, \dots, \underline{x}^{(m)} \in \mathbb{R}^n$. Dann können wir die obige Lernregel als schrittweises Verfahren einer allgemeineren Regel ansehen, welche die Gewichte zu allen Eingabevektoren *gleichzeitig* anpaßt, anstatt dies nach jedem einzelnen Vektor zu tun. Aufgrund von $y = \underline{x}^T \underline{w}$ erhalten wir dann:

$$\begin{aligned} \underline{w} &\leftarrow \underline{w} + \gamma \cdot \sum_{i=1}^m \underline{x}^{(i)} \cdot \underline{x}^{(i)T} \underline{w} \\ \underline{w} &\leftarrow \underline{w} \cdot \frac{1}{\|\underline{w}\|_2} \end{aligned}$$

Die Lernrate γ können wir nun schreiben als $\gamma = \gamma' \cdot m \cdot \frac{1}{m}$ und m (da γ eine beliebige reelle Zahl ist) in γ' hineinziehen (im Prinzip erhalten wir eine neue Variable γ' , aber um die Notation einfach zu halten, nennen wir sie einfach auch wieder γ). Dadurch wird der Summenausdruck zu einem Erwartungswert und wir erhalten abschließend die Lernregel:

$$\begin{aligned} \underline{w} &\leftarrow \underline{w} + \gamma \cdot \langle \underline{x} \cdot \underline{x}^T \rangle \cdot \underline{w} \\ \underline{w} &\leftarrow \underline{w} \cdot \frac{1}{\|\underline{w}\|_2} \end{aligned}$$

Was ist aber nun das Ziel dieser Lernregel? Um dies herauszufinden, können

wir folgendermaßen vorgehen. Wir können sagen, daß das Ziel der Iteration erreicht ist, wenn sich \underline{w} in weiteren Iterationsschritten nicht mehr verändert, \underline{w} also zu einem stabilen Zustand konvergiert ist. Aufgrund der Normierung von \underline{w} ist dies dann der Fall, wenn $\underline{w} + \gamma \cdot \langle \underline{x} \cdot \underline{x}^T \rangle \cdot \underline{w}$ wieder ein Vielfaches von \underline{w} ist. Setzen wir $C_{xx} := \langle \underline{x} \cdot \underline{x}^T \rangle$, dann gilt im Falle der Konvergenz also:

$$\begin{aligned} \underline{w} + \gamma \cdot C_{xx} \cdot \underline{w} &= \alpha \cdot \underline{w} \\ \iff C_{xx} \cdot \underline{w} &= \frac{\alpha-1}{\gamma} \cdot \underline{w} \end{aligned}$$

Wir sehen nun, daß \underline{w} im Falle der Konvergenz ein Eigenvektor der Matrix C_{xx} ist. Zu klären bleibt noch, ob und wohin genau \underline{w} nun tatsächlich konvergiert. Schauen wir uns dazu einmal die Eigenvektoren $\underline{e}_1, \dots, \underline{e}_n$ und die zugehörigen Eigenwerte $\lambda_1, \dots, \lambda_n$ dieser Matrix an, wobei $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ gelte. Nach einem Lernschritt gilt dann für den neuen Gewichtsvektor \underline{w}' und einen beliebigen Eigenvektor \underline{e}_i :

$$\begin{aligned} \underline{w}'^T \cdot \underline{e}_i &= \frac{1}{\|\underline{w} + \gamma \cdot C_{xx} \cdot \underline{w}\|_2} \cdot (\underline{w} + \gamma \cdot C_{xx} \cdot \underline{w})^T \cdot \underline{e}_i \\ &= \frac{1}{\|\underline{w} + \gamma \cdot C_{xx} \cdot \underline{w}\|_2} \cdot (\underline{w}^T + \gamma \cdot \underline{w}^T \cdot C_{xx}) \cdot \underline{e}_i \\ &= \frac{1}{\|\underline{w} + \gamma \cdot C_{xx} \cdot \underline{w}\|_2} \cdot (\underline{w}^T \cdot \underline{e}_i + \gamma \cdot \underline{w}^T \cdot \lambda_i \cdot \underline{e}_i) \\ &= \frac{1}{\|\underline{w} + \gamma \cdot C_{xx} \cdot \underline{w}\|_2} \cdot (1 + \gamma \lambda_i) \cdot \underline{w}^T \underline{e}_i \\ &\geq \frac{1}{\|\underline{w}\|_2 + \|\gamma \cdot C_{xx} \cdot \underline{w}\|_2} \cdot (1 + \gamma \lambda_i) \cdot \underline{w}^T \underline{e}_i \\ &= \frac{1}{1 + \gamma \cdot \|C_{xx} \cdot \underline{w}\|_2} \cdot (1 + \gamma \lambda_i) \cdot \underline{w}^T \underline{e}_i \\ &> \frac{1}{1 + \gamma \lambda_1} \cdot (1 + \gamma \lambda_i) \cdot \underline{w}^T \underline{e}_i \end{aligned}$$

Die letzte Zeile gilt dabei unter der Bedingung, daß \underline{w} nicht bereits schon der Eigenvektor von C_{xx} mit größtem Eigenwert ist und folgt aus der Tatsache, daß die Länge des Vektors $C_{xx} \cdot \underline{w}$ für $\|\underline{w}\| = 1$ höchstens so groß ist, wie der maximale Eigenwert von C_{xx} (man kann zeigen, daß $\max_{\underline{v}, \|\underline{v}\|_2=1} \|C_{xx} \cdot \underline{v}\| = \lambda_1$). Insbesondere folgt für \underline{e}_1 , den Eigenvektor mit größtem Eigenwert:

$$\underline{w}'^T \cdot \underline{e}_1 = \alpha \cdot \underline{w}^T \cdot \underline{e}_1, \quad \alpha := \frac{1 + \gamma \lambda_1}{1 + \gamma \|C_{xx} \cdot \underline{w}\|_2} > 1$$

Da \underline{w} nach jedem Iterationsschritt normiert wird und das Skalarprodukt zwischen

\underline{w} und \underline{e}_1 zunimmt, bedeutet dies, daß \underline{w} zunehmend in die Richtung von \underline{e}_1 rotiert wird. α bestimmt dabei die Geschwindigkeit des Konvergenzvorganges. Da $\alpha > 1$, ist nachgewiesen, daß die obige Iteration *immer* zu einem Eigenvektor von C_{xx} mit größtem Eigenwert konvergiert, und das unabhängig von der Lernrate γ !

nach oben

Zusammenhang zur Dekorrelation

Was hat dies nun mit Dekorrelation zu tun? Nehmen wir dazu einmal an, wir haben nicht nur den Eigenvektor mit größtem Eigenwert, sondern sämtliche Eigenvektoren von C_{xx} in den Gewichtsvektoren $\underline{w}_1, \dots, \underline{w}_n$ von n Neuronen gelernt, die alle lediglich eine lineare Ausgabefunktion der Form $S(z) = z$ besitzen. Dann berechnet sich die Ausgabe y_i des i -ten Neurons bei Eingabe eines \underline{x} als $y_i = \underline{x}^T \underline{w}_i$. Nehmen wir nun kurz einmal an, es existieren lineare Abhängigkeiten zwischen einem beliebigen y_i und einem y_j mit $i \neq j$. Dann folgt, mit zwei reellen Zahlen $a, b \in \mathbb{R}, a \neq 0$:

$$\begin{aligned} y_i &= ay_j + b \\ \Rightarrow \langle (y_i - \langle y_i \rangle)(y_j - \langle y_j \rangle) \rangle &= \langle (ay_j + b - \langle ay_j + b \rangle)(y_j - \langle y_j \rangle) \rangle \\ &= a \langle (y_j - \langle y_j \rangle)(y_j - \langle y_j \rangle) \rangle \\ &= a \langle (y_j - \langle y_j \rangle)^2 \rangle \\ &\neq 0 \end{aligned}$$

Den Ausdruck $\langle (y_i - \langle y_i \rangle)(y_j - \langle y_j \rangle) \rangle$ bezeichnet man dabei als Kovarianz von y_i und y_j . Wie wir gesehen haben, ist sie ungleich Null, falls y_i und y_j linear abhängig (bzw. korreliert) sind. Es gilt nun:

$$\begin{aligned}
\text{cov}(y_i, y_j) &= \langle (y_i - \langle y_i \rangle)(y_j - \langle y_j \rangle) \rangle \\
&= \langle \underline{w}_i^T (\underline{x} - \langle \underline{x} \rangle) \underline{w}_j^T (\underline{x} - \langle \underline{x} \rangle) \rangle \\
&= \underline{w}_i^T \langle (\underline{x} - \langle \underline{x} \rangle)(\underline{x} - \langle \underline{x} \rangle)^T \rangle \underline{w}_j \\
&= \underline{w}_i^T \langle \underline{x} \underline{x}^T - \underline{x} \langle \underline{x} \rangle^T - \langle \underline{x} \rangle \underline{x}^T + \langle \underline{x} \rangle \langle \underline{x} \rangle^T \rangle \underline{w}_j \\
&= \underline{w}_i^T \left(\langle \underline{x} \underline{x}^T \rangle - \langle \underline{x} \langle \underline{x} \rangle^T \rangle - \langle \langle \underline{x} \rangle \underline{x}^T \rangle + \langle \langle \underline{x} \rangle \langle \underline{x} \rangle^T \rangle \right) \underline{w}_j \\
&= \underline{w}_i^T \langle \underline{x} \underline{x}^T \rangle \underline{w}_j - \underline{w}_i^T \langle \underline{x} \rangle \langle \underline{x} \rangle^T \underline{w}_j \\
&= \underline{w}_i^T C_{xx} \underline{w}_j - (\underline{w}_i^T \langle \underline{x} \rangle)^2 \\
&= \underline{w}_i^T \lambda_j \underline{w}_j - (\underline{w}_i^T \langle \underline{x} \rangle)^2 \\
&= 0 - (\underline{w}_i^T \langle \underline{x} \rangle)^2
\end{aligned}$$

Die letzte Zeile folgt dabei aus der Tatsache, daß die Eigenvektoren von C_{xx} orthogonal zueinander sind. Wie wir sehen, wird die Kovarianz Null, falls der Erwartungswert $\langle \underline{x} \rangle$ der Eingabevektoren gerade der Nullvektor ist. Die Matrix C_a wird dann auch als die *Autokorrelationsmatrix* der Eingabevektoren bezeichnet (die genau Definition der Autokorrelationsmatrix ist $C_{xx} := \langle (\underline{x} - \langle \underline{x} \rangle)(\underline{x} - \langle \underline{x} \rangle)^T \rangle$). Eine wichtige Voraussetzung besteht also darin, daß die Daten zuvor auf Erwartungswert Null zentriert wurden! Ist dies der Fall, so haben wir gezeigt, daß ein Netz bestehend aus n Neuronen, deren Gewichtsvektoren den Eigenvektoren der Autokorrelationsmatrix entsprechen, in der Tat dekorrelierte Ausgaben produziert.

nach oben

Lernen der übrigen Gewichtsvektoren

Was nun noch fehlt, ist ein Verfahren, wie wir neben dem Eigenvektor mit größtem Eigenwert zusätzlich noch die übrigen Eigenvektoren lernen lassen können.

Glücklicherweise läßt sich die obige Iteration zum Lernen des Eigenvektors mit größtem Eigenwert auch dazu verwenden, die übrigen Eigenvektoren zu lernen. Alles was wir dazu tun müssen, ist den Eigenvektor mit größtem Eigenwert aus der Autokorrelationsmatrix zu entfernen, ohne die übrigen Eigenvektoren zu beeinflussen. Anschließend können wir unsere Iteration anwenden, um den

Eigenvektor mit nächstgrößerem Eigenwert zu finden (der ja nun den größten Eigenwert bezüglich der neuen Matrix besitzt). Dazu gehen wir folgendermaßen vor. Nachdem wir den ersten Eigenvektor - nennen wir ihn \underline{w}_1 - gelernt haben, bilden wir neue Eingabevektoren \underline{x}' , nach dem Schema:

$$\underline{x}' = \underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1$$

Ist \underline{e}_i ein beliebiger von \underline{w}_1 verschiedener Eigenvektor, dann hat die resultierende

Autokorrelationsmatrix $C'_{xx} = \langle \underline{x}' \underline{x}'^T \rangle$ die Eigenschaft:

$$\begin{aligned} C'_{xx} \underline{w}_1 &= \langle \underline{x}' \underline{x}'^T \rangle \underline{w}_1 \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)(\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)^T \rangle \underline{w}_1 \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)(\underline{x}^T \underline{w}_1 - \underline{w}_1^T \underline{x} \cdot \underline{w}_1^T \underline{w}_1) \rangle \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)(\underline{x}^T \underline{w}_1 - \underline{w}_1^T \underline{x}) \rangle \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1) \cdot 0 \rangle \\ &= 0 \end{aligned}$$

Sowie:

$$\begin{aligned} C'_{xx} \underline{e}_i &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)(\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)^T \rangle \underline{e}_i \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1)(\underline{x}^T \underline{e}_i - \underbrace{\underline{w}_1^T \underline{x} \cdot \underline{w}_1^T \underline{e}_i}_{=0}) \rangle \\ &= \langle (\underline{x} - \underline{w}_1^T \underline{x} \cdot \underline{w}_1) \cdot \underline{x}^T \underline{e}_i \rangle \\ &= \langle \underline{x} \underline{x}^T \underline{e}_i - \underline{w}_1^T \underline{x} \cdot \underline{w}_1 \cdot \underline{x}^T \underline{e}_i \rangle \\ &= C_{xx} \cdot \underline{e}_i - \underline{w}_1^T \cdot C_{xx} \cdot \underline{e}_i \cdot \underline{w}_1 \\ &= C_{xx} \cdot \underline{e}_i - \lambda_i \cdot \underbrace{\underline{w}_1^T \cdot \underline{e}_i \cdot \underline{w}_1}_{=0} \\ &= C_{xx} \underline{e}_i \end{aligned}$$

Die neue Autokorrelationsmatrix besitzt also alle Eigenvektoren und zugehörigen Eigenwerte der ursprünglichen Matrix, mit Ausnahme des Eigenvektors \underline{w}_1 , den wir zuvor gelernt haben. Das Verfahren läßt sich dann von neuem anwenden. Zusammenfassend erhalten wir dann die folgenden

Lernregeln (startend bei $i = 1$ für den Eigenvektor mit größtem Eigenwert):

Iteriere, bis Fehler klein genug :

$$\underline{w}_i \leftarrow \underline{w}_i + \gamma \cdot \langle \underline{x} \cdot \underline{x}^T \rangle \cdot \underline{w}_i$$

$$\underline{w}_i \leftarrow \underline{w}_i \cdot \frac{1}{\|\underline{w}_i\|_2}$$

Berechne :

$$\forall \underline{x} : \underline{x} \leftarrow \underline{x} - \underline{w}_i^T \underline{x} \cdot \underline{w}_i$$

Nächster Schritt :

$$i \leftarrow i + 1$$

Das zu dieser Lernregel gehörende Netz ist in Abbildung 2 dargestellt und wird als *Sangernetz* bezeichnet. Das Training erfolgt hier schichtweise, startend mit zufälligen normierten Gewichtsvektoren. Sobald eine Schicht ausreichend trainiert wurde, wird das Training mit der nachfolgenden Schicht fortgesetzt. Wird dem Netz nach Abschluß des Trainings dann ein Eingabevektor präsentiert und die Netzausgabe berechnet, so ist diese dekorreliert.

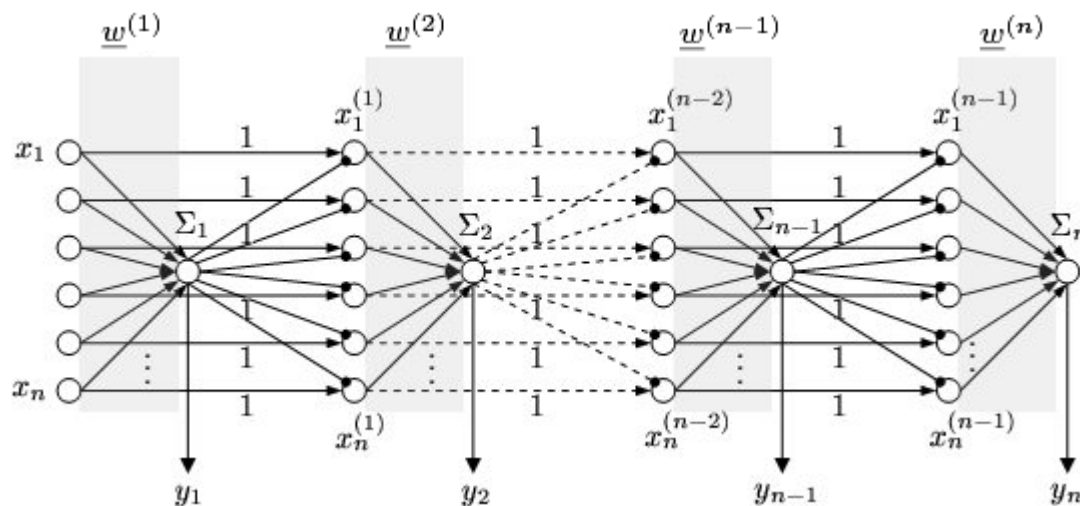


Abbildung 2. Struktur eines PCA-Netzes (*Sangernetz*). Die Eingabe des Netzes ist der Vektor $\underline{x} = [x_1, \dots, x_n]^T$. Das folgende Neuron (Σ_1) besitzt zur Eingabe x_i das Verbindungsgewicht $w_i^{(1)}$ und bildet dann seine Ausgabe als $y_1 = \underline{x}^T \underline{w}^{(1)}$ mit

$\underline{w}^{(1)} = [w_1^{(1)}, \dots, w_n^{(1)}]^T$. Das i -te Neuron der $\underline{x}^{(1)}$ -Schicht erhält dann y_1 und x_i als Eingabe, wobei die Kreise an den Neuronen andeuten sollen, daß die Ausgabe von Σ_1 einen hemmenden Einfluß auf das betreffende Neuron ausübt. Das

Verbindungsgewicht zwischen $x_i^{(1)}$ und Σ_1 soll dabei gerade auch wieder $w_i^{(1)}$ entsprechen. Die Ausgabe dieser Neuronen berechnet sich dann zu $x_i - y_1 \cdot w_i^{(1)}$ (alternativ kann auch angenommen werden, daß das Gewicht zwischen Σ_1 und $x_i^{(1)}$ gerade $-w_i^{(1)}$ beträgt und die Ausgabe wie üblich durch gewichtete Summation erfolgt). Die Ausgaben der Neuronen der $\underline{x}^{(1)}$ -Schicht bilden dann die Eingabe für das Σ_2 -Neuron, usw. Die Ausgabe der einzelnen Σ -Neuronen ist dann die Ausgabe des Netzes.

Dazu ist noch anzumerken, daß nach erfolgreichem Training aller Schichten der Netzausgabevektor $\underline{y} = [y_1, \dots, y_n]^T$ zu einer Eingabe \underline{x} sich direkt als $\underline{y} = W^T$ berechnet, wobei $W = [\underline{w}_1, \dots, \underline{w}_n]$ die Matrix der Gewichtsvektoren ist. Eine Berechnung der x -Eingaben für jedes einzelne Neuron der Folgeschichten ist also nach abgeschlossenem Training nicht mehr erforderlich. Dies hängt damit zusammen, daß für $i \neq j$ die Beziehung $(\underline{x} - \underline{w}_i^T \underline{x} \cdot \underline{w}_i)^T \cdot \underline{w}_j = \underline{x}^T \underline{w}_j$ gilt, da die \underline{w} als Eigenvektoren der Autokorrelationsmatrix paarweise orthogonal zueinander sind. Das Netz verhält sich dann rein linear und berechnet eine lineare Transformation, die durch die Matrix W gegeben ist. Da W aus orthogonalen und normierten Spaltenvektoren besteht, also eine orthonormale Matrix ist, führt W eine Rotation im Eingaberaum aus. Diese läßt sich auch wieder umkehren und es gilt die Beziehung $\underline{x} = W \underline{y}$, so daß sich die transformierten Eingabevektoren wieder zurückrechnen lassen.

Eine recht nützliche Eigenschaft des Sangernetzes ist die Tatsache, daß aufgrund seiner Konstruktion, wie wir sie oben kennengelernt haben, die Koeffizienten y_1, \dots, y_n absteigend nach der Größe der Eigenwerte der Eigenvektoren, zu denen sie gehören, geordnet sind. Die Eigenwerte entsprechen dabei der Varianz in Raumrichtung des entsprechenden Eigenvektors, so daß die y_i gewissermaßen absteigend nach der Relevanz geordnet sind, die sie bezüglich einer Rekonstruktion des ursprünglichen Vektors besitzen.

nach oben

Konvergenzkriterium

Abschließend sollte vielleicht noch erwähnt werden, welches Kriterium mit "Der Fehler ist klein genug" in der Lernregel oben gemeint ist. Nun, wir können die Iteration für einen Gewichtsvektor \underline{w}_i abbrechen, wenn er einen Eigenvektor der Autokorrelationsmatrix darstellt (oder zumindest sehr nahe dran ist). Da wir sowieso den Ausdruck $C_{xx}\underline{w}_i$ berechnen müssen (um während der Iteration den Gewichtsvektor \underline{w}_i anzupassen) und dieser gegen $\lambda_i \underline{w}_i$ konvergiert, konvergiert $\frac{C_{xx}\underline{w}_i}{\|C_{xx}\underline{w}_i\|_2}$ gegen \underline{w}_i , so daß

$$\delta := \left\| \frac{C_{xx}\underline{w}_i}{\|C_{xx}\underline{w}_i\|_2} - \underline{w}_i \right\|_2$$

gegen Null geht und damit ein gutes Konvergenzkriterium darstellt. Sehr wichtig zu erwähnen ist dabei allerdings noch, daß das Sangernetz keine Eigenvektoren zu Eigenwert 0 berechnet. Allerdings können solche Eigenwerte identifiziert werden, da der Ausdruck $\|C_{xx}\underline{w}_i\|_2$ in diesem Fall gegen Null geht und die Autokorrelationsmatrix (gebildet aus den modifizierten Eingabevektoren für die i-te Schicht) sich der Nullmatrix nähert.

nach oben