

*IBM SPSS Neural Networks 29*



**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 19 gelesen werden.

**Produktinformation**

Diese Ausgabe bezieht sich auf Version 29, Release 0, Modifikation 1 von IBM® SPSS Statistics und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

© **Copyright International Business Machines Corporation .**

---

# Inhaltsverzeichnis

<b>Kapitel 1. Neuronale Netze.....</b>	<b>1</b>
Einführung in Neural Networks (Neuronale Netze).....	1
Was ist ein neuronales Netz?.....	1
Struktur neuronaler Netze.....	2
Mehrschichtiges Perzeptron.....	3
Partitionen .....	5
Architektur .....	6
Training .....	7
Ausgabe .....	9
Speichern .....	10
Export .....	11
Optionen .....	11
Radiale Basisfunktion.....	12
Partitionen .....	14
Architektur .....	14
Ausgabe .....	15
Speichern .....	16
Export .....	17
Optionen .....	17
<b>Bemerkungen.....</b>	<b>19</b>
Marken.....	20
<b>Index.....</b>	<b>23</b>



---

# Kapitel 1. Neuronale Netze

Die folgenden Funktionen für neuronale Netze sind in SPSS Statistics Premium Edition oder der Option "Neural Networks" enthalten.

---

## Einführung in Neural Networks (Neuronale Netze)

---

Neuronale Netze sind aufgrund ihrer Leistungsfähigkeit, Flexibilität und Benutzerfreundlichkeit das bevorzugte Tool für zahlreiche Anwendungen auf dem Gebiet des prädiktiven Data-Mining. Prädiktive neuronale Netze sind besonders nützlich bei Anwendungen, denen ein komplexer Prozess zugrunde liegt, wie beispielsweise:

- Vorhersage der Verbrauchernachfrage zur Rationalisierung von Produktions- und Lieferkosten.
- Vorhersage der Antwortwahrscheinlichkeit bei Marketingaktionen mit Postsendungen, um zu ermitteln, an welche Haushalte im Verteiler ein Angebot gesendet werden sollte.
- Scoring eines Antragstellers, um dessen Kreditrisiko zu ermitteln.
- Aufdecken betrügerischer Transaktionen in einer Datenbank mit Versicherungsforderungen.

Neuronale Netze, die in Vorhersageanwendungen verwendet werden, wie z. B. MLP-Netze (Multilayer Perzeptron) und RBF-Netze (radiale Basisfunktion), werden in dem Sinne überwacht, dass die modellvorhergesagten Ergebnisse mit bekannten Werten der Zielvariablen verglichen werden können. Mit der Option Neural Networks können Sie MLP- und RBF-Netze anpassen und die so entstehenden Modelle für das Scoring speichern.

## Was ist ein neuronales Netz?

Der Begriff **neuronales Netz** bezieht sich auf eine locker miteinander verwandte Modellfamilie, die durch einen großen Parameterraum und eine flexible Struktur gekennzeichnet ist, die sich aus den Studien zur Funktionsweise des Gehirns herleitet. Als die Modellfamilie wuchs, wurden die meisten neuen Modelle für Anwendungen außerhalb der Biologie entwickelt, obwohl ein Großteil der zugehörigen Terminologie noch die Ursprünge erkennen lässt.

Die spezifischen Definitionen für neuronale Netze sind so vielfältig wie ihre Einsatzgebiete. Da bisher keine einzelne Definition die gesamte Modellfamilie korrekt abdeckt, sollten Sie die folgende Beschreibung beachten<sup>1</sup>:

Ein neuronales Netz ist ein verteilter massiv-paralleler Prozessor mit einer natürlichen Neigung zur Speicherung von experimentellem Wissen und seiner Bereitstellung. Es ähnelt dem Hirn in zwei Aspekten:

- Wissen wird vom Netz durch einen Lernprozess erworben.
- Interneuronale Verbindungsstärken, auch als synaptische Gewichtungen bekannt, dienen zum Speichern des Wissens.

Für eine Diskussion darüber, warum diese Definition vielleicht zu restriktiv ist, siehe<sup>2</sup>.

Wenn wir neuronale Netze mit dieser Definition von traditionellen statistischen Methoden unterscheiden möchten, ist das, was *nicht* gesagt wurde, ebenso bedeutsam, wie der Text der Definition selbst. So kann beispielsweise das traditionelle lineare Regressionsmodell Wissen durch die Methode der kleinsten Quadrate erwerben und dieses Wissen in den Regressionskoeffizienten speichern. In dieser Hinsicht handelt es sich dabei um ein neuronales Netz. In der Tat lässt sich die Auffassung vertreten, dass die lineare Regression einen Sonderfall bestimmter neuronaler Netze darstellt. Allerdings weist die lineare Regression eine starre Modellstruktur und ein Set von Annahmen auf, die angewendet werden, bevor aus den Daten "gelernt" wird.

---

<sup>1</sup> Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation*, 2. Aufl. New York: Macmillan College Publishing.

<sup>2</sup> Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Im Gegensatz dazu stellt die oben angegebene Definition nur minimale Anforderungen an Struktur und Annahmen. Daher kann ein neuronales Netz eine Annäherung an eine große Bandbreite statistischer Modelle bieten, ohne dass von vornherein Hypothesen über bestimmte Beziehungen zwischen den abhängigen und den unabhängigen Variablen erforderlich sind. Stattdessen wird die Form der Beziehungen im Laufe des Lernprozesses bestimmt. Wenn eine lineare Beziehung zwischen abhängigen und unabhängigen Variablen angemessen ist, sollten die Ergebnisse des neuronalen Netzes eine große Ähnlichkeit zu denen des linearen Regressionsmodells darstellen. Wenn eine nicht lineare Beziehung angemessener ist, ähnelt das neuronale Netz automatisch der "richtigen" Modellstruktur.

Der Preis für diese Flexibilität besteht darin, dass die synaptischen Gewichtungen eines neuronalen Netzes nicht leicht zu interpretieren sind. Wenn Sie also versuchen, den zugrunde liegenden Prozess zu erklären, der zu den Beziehungen zwischen den abhängigen und den unabhängigen Variablen führt, sollten Sie lieber ein traditionelleres statistisches Modell verwenden. Wenn jedoch die Interpretierbarkeit des Modells nicht von Belang ist, können Sie häufig schneller mithilfe eines neuronalen Netzes zu guten Modellergebnissen kommen.

## Struktur neuronaler Netze

Auch wenn neuronale Netze nur minimale Anforderungen an die Modellstruktur und die geltenden Annahmen stellen, ist es dennoch nützlich, einen Einblick in die allgemeine **Netzarchitektur** zu haben. Beim MLP- oder RBF-Netz handelt es sich um eine Funktion von Prädiktoren (auch als Eingaben oder unabhängige Variablen bezeichnet), die den Vorhersagefehler der Zielvariablen (auch als Ausgaben bezeichnet) minimiert.

Betrachten Sie das Dataset *bankloan.sav*, das im Lieferumfang des Produkts enthalten ist. In diesem Dataset sollen aus einem Pool von Kreditantragstellern die Personen ermittelt werden, die mit großer Wahrscheinlichkeit zahlungsunfähig werden. Bei einem auf dieses Problem angewendeten MLP- oder RBF-Netz handelt es sich um eine Funktion von Messungen, die den Fehler bei der Vorhersage der Zahlungsunfähigkeit minimiert. Folgende Abbildung dient zur Angabe der Form dieser Funktion.

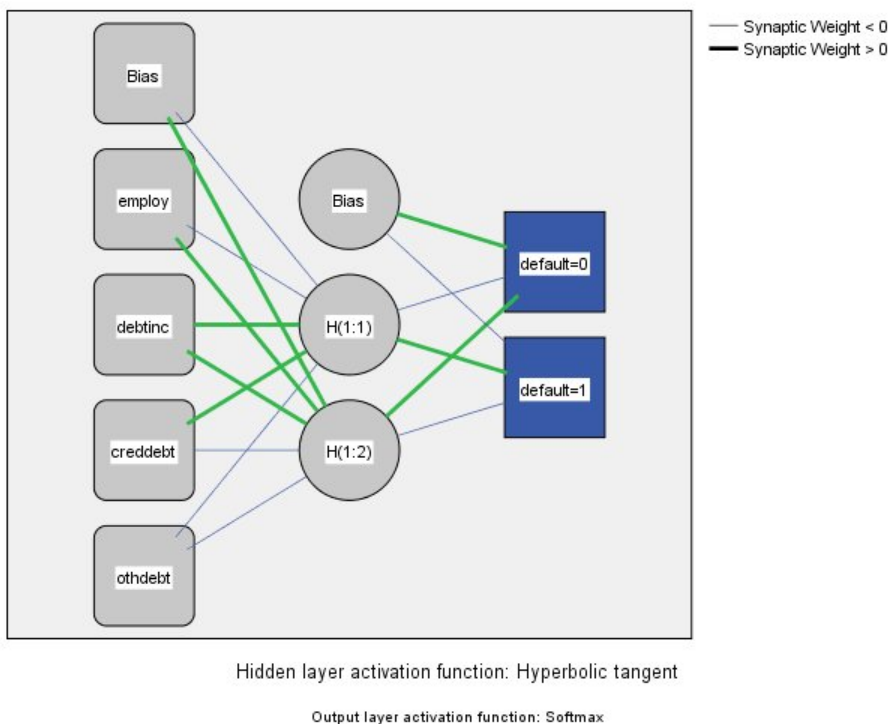


Abbildung 1. Feedforward-Architektur mit einer verborgenen Schicht

Diese Struktur ist als **Feedforward-Architektur** bekannt, da die Verbindungen im Netz ohne Rückkopplungsschleifen vorwärts von der Eingabeschicht zur Ausgabeschicht verlaufen. In dieser Abbildung gilt:

- Die **Eingabeschicht** enthält die Prädiktoren.

- Die **verborgene Schicht** enthält nicht sichtbare Knoten (Einheiten). Der Wert jeder verdeckten Einheit ist eine Funktion der Prädiktoren; die genaue Form der Funktion hängt zum Teil vom Netztyp und zum Teil von den vom Benutzer kontrollierbaren Spezifikationen ab.
- Die **Ausgabeschicht** enthält die Antworten. Da es sich bei den früheren Fällen von Zahlungsverzug um eine kategoriale Variable mit zwei Kategorien handelt, wird sie als zwei Indikatorvariablen umcodiert. Jede Ausgabeeinheit ist jeweils eine Funktion der verborgenen Einheiten. Auch hier hängt die genaue Form der Funktion zum Teil vom Netztyp und zum Teil von den vom Benutzer festlegbaren Spezifikationen ab.

Beim MLP-Netz ist eine zweite verborgene Schicht zulässig. In diesem Fall ist jede Einheit der zweiten verborgenen Schicht eine Funktion der Einheiten in der ersten verborgenen Schicht und jede Antwort ist eine Funktion der Einheiten in der zweiten verborgenen Schicht.

## Mehrschichtiges Perzeptron

Die Prozedur "Mehrschichtiges Perzeptron" (Multilayer Perceptron, MLP) erstellt ein Vorhersagemodell für mindestens eine abhängige Variable (Zielvariable), das auf den Werten der Prädiktorvariablen beruht.

**Beispiele.** Im Folgenden finden Sie zwei Szenarios, die die Prozedur MLP verwenden:

Eine Kreditsachbearbeiterin in einer Bank muss in der Lage sein, Merkmale zu ermitteln, die auf Personen hindeuten, die mit hoher Wahrscheinlichkeit ihre Kredite nicht zurückzahlen, und diese Merkmale zur Feststellung eines guten bzw. schlechten Kreditrisikos einzusetzen. Mit einer Stichprobe vorheriger Kunden kann sie ein mehrschichtiges Perzeptron trainieren, die Analyse mithilfe einer Holdout-Stichprobe vorheriger Kunden validieren und dann das Netz nutzen, um potenzielle Kunden als gute oder schlechte Kreditrisiken einzustufen.

Ein Krankenhaussystem möchte die Kosten und die Aufenthaltsdauer für Patienten aufzeichnen, die zur Behandlung eines Herzinfarkts aufgenommen wurden. Durch genaue Schätzungen dieser Messwerte kann die Krankenhausverwaltung die verfügbare Bettenkapazität während der Behandlung der Patienten besser verwalten. Mithilfe der Behandlungsdatensätze einer Stichprobe von Patienten, die eine Behandlung für MI erhalten haben, kann der Administrator ein Netzwerk ausbilden, um die Kosten und die Dauer des Aufenthalts vorherzusagen.












Erläuterung der Daten

**Abhängige Variablen.** Die abhängigen Variablen können wie folgt gestaltet sein:

- *Nominal.* Eine Variable kann als nominal behandelt werden, wenn ihre Werte Kategorien ohne interne Rangfolge darstellen (z. B. die Abteilung des Unternehmens, in der ein Mitarbeiter arbeitet). Beispiele für nominale Variablen sind Region, Postleitzahl oder Religionszugehörigkeit.
- *Ordinal.* Eine Variable kann als ordinal behandelt werden, wenn ihre Werte Kategorien mit einer intrinsischen Rangfolge darstellen (z. B. Grad der Servicezufriedenheit von hoch unzufrieden bis hoch zufrieden). Ordinale Variablen treten beispielsweise bei Einstellungsmessungen (Zufriedenheit oder Vertrauen) und bei Präferenzbeurteilungen auf.
- *Skala.* Eine Variable kann als metrisch (stetig) behandelt werden, wenn ihre Werte geordnete Kategorien mit einer aussagekräftigen Metrik darstellen, sodass Abstandsvergleiche zwischen Werten angemessen sind. Metrische Variablen sind beispielsweise Alter (in Jahren) oder Einkommen (in Geldeinheiten).

Bei der Prozedur wird davon ausgegangen, dass allen abhängigen Variablen das richtige Messniveau zugewiesen wurde. Sie können das Messniveau für eine Variable jedoch vorübergehend ändern. Klicken Sie hierzu mit der rechten Maustaste auf die Variable in der Liste der Quellvariablen und wählen Sie das gewünschte Messniveau im Popup-Menü aus.

Messniveau und Datentyp sind durch ein Symbol neben der jeweiligen Variablen in der Variablenliste gekennzeichnet:

	Numerisch	Zeichenfolge	Datum	Zeit
Metrisch (stetig)		nicht zutreffend		
Ordinalzahl				
Nominal				

**Prädiktorvariablen.** Prädiktoren können als Faktoren (kategorial) oder als Kovariaten (metrisch) angegeben werden.

**Codierung für kategoriale Variablen.** Die Prozedur codiert vorübergehend für die Dauer ihrer Ausführung kategoriale Prädiktoren und abhängige Variablen mithilfe der "Eins-aus-c"-Codierung neu. Wenn es  $c$  Kategorien für eine Variable gibt, wird die Variable als  $c$  Vektoren gespeichert. Dabei wird die erste Kategorie als  $(1,0,\dots,0)$  angegeben, die zweite Kategorie als  $(0,1,0,\dots,0),\dots$  und die letzte Kategorie als  $(0,0,\dots,0,1)$ .

Dieses Codierungsschema erhöht die Anzahl der synaptischen Gewichtungen und kann zu einer Verlangsamung des Trainings führen, "kompaktere" Codierungsmethoden führen jedoch in der Regel zu neuronalen Netzen mit geringer Anpassungsgüte. Wenn Ihr Netztraining sehr langsam voranschreitet, versuchen Sie, die Anzahl der Kategorien in Ihren kategorialen Prädiktoren zu reduzieren, indem Sie ähnliche Kategorien kombinieren oder Fälle mit extrem seltenen Kategorien löschen.

Jede "Eins-aus-c"-Codierung beruht auf den Trainingsdaten, selbst wenn eine Test- oder Holdout-Stichprobe definiert wurde (siehe „Partitionen“ auf Seite 5). Wenn also die Test- oder Holdout-Stichproben Fälle mit Prädiktorkategorien enthalten, die in den Trainingsdaten nicht vorhanden sind, werden diese Fälle nicht in der Prozedur oder beim Scoring verwendet. Wenn die Test- oder Holdout-Stichproben Fälle mit Kategorien abhängiger Variablen enthalten, die in den Trainingsdaten nicht vorhanden sind, werden diese Fälle zwar nicht in der Prozedur, jedoch möglicherweise beim Scoring verwendet.

**Neuskalierung.** Metrische abhängige Variablen und Kovariaten werden standardmäßig neu skaliert, um das Training des Netzes zu verbessern. Jede Neuskalierung beruht auf den Trainingsdaten, selbst wenn eine Test- oder Holdout-Stichprobe definiert wurde (siehe „Partitionen“ auf Seite 5). Das bedeutet, dass je nach Neuskalierungstyp Mittelwert, Standardabweichung, Mindestwert oder Maximalwert einer Kovariaten oder abhängigen Variablen ausschließlich anhand der Trainingsdaten berechnet wird. Wenn Sie eine Variable zur Festlegung von Partitionen angeben, müssen diese Kovariaten oder abhängigen Variablen in der Trainings-, Test- und Holdout-Stichprobe ähnliche Verteilungen aufweisen.

**Häufigkeitsgewichtungen.** Häufigkeitsgewichtungen werden von dieser Prozedur ignoriert.

**Reproduzieren der Ergebnisse.** Wenn Sie Ihre Ergebnisse exakt reproduzieren möchten, müssen Sie nicht nur dieselben Einstellungen für die Prozedur, sondern auch denselben Initialisierungswert für den Zufallszahlengenerator, dieselbe Datenreihenfolge und dieselbe Variablenreihenfolge verwenden. Weitere Details zu diesem Problem folgen:

- **Generierung von Zufallszahlen.** Die Prozedur verwendet Zufallszahlengenerierung während der Zufallszuweisung von Partitionen, zufällige Ziehung von Teilstichproben für die Initialisierung der synaptischen Gewichtungen, zufällige Ziehung von Teilstichproben für die automatische Architekturauswahl und den Algorithmus der simulierten Abkühlung für die Initialisierung der Gewichtungen und die automatische Architekturauswahl. Um zu einem späteren Zeitpunkt dieselben randomisierten Ergebnisse zu reproduzieren, müssen Sie vor jeder Ausführung der Prozedur "Mehrschichtiges Perzeptron" denselben Initialisierungswert für den Zufallszahlengenerator verwenden. .
- **Fallreihenfolge.** Die Trainingsmethoden "Online" und "Mini-Batch" (siehe „Training“ auf Seite 7) sind explizit von der Fallreihenfolge abhängig. Allerdings ist sogar Batch-Training von der Fallreihenfolge



abhängig, da die Initialisierung der synaptischen Gewichtungen die Ziehung einer Teilstichprobe aus dem Dataset beinhaltet.

Um die Auswirkungen der Reihenfolge zu minimieren, mischen Sie die Fälle in zufälliger Reihenfolge. Prüfen Sie daher die Stabilität einer bestimmten Lösung, indem Sie verschiedene Lösungen abrufen, bei denen die Fälle in einer unterschiedlichen, zufällig ausgewählten Reihenfolge sortiert sind. In Situationen mit extrem umfangreichen Dateien können mehrere Durchgänge mit jeweils einer Stichprobe von Fällen durchgeführt werden, die in unterschiedlicher, zufällig ausgewählter Reihenfolge sortiert ist.

- **Reihenfolge der Variablen.** Die Ergebnisse können von der Reihenfolge der Variablen in der Faktorenliste und der Kovariatenliste beeinflusst werden, da die zugewiesenen Anfangswerte ein anderes Muster aufweisen, wenn die Reihenfolge der Variablen geändert wird. Wie bei den Effekten der Fallreihenfolge können Sie auch eine andere Reihenfolge der Variablen ausprobieren (durch Ziehen und Ablegen in der Liste der Faktoren oder Kovariaten), um die Stabilität einer bestimmten Lösung einzuschätzen.

Erstellen eines Netzes mit mehrschichtigen Perzeptronen

Wählen Sie die folgenden Befehle aus den Menüs aus:

### **Analysieren > Neuronale Netze > Mehrschichtiges Perzeptron...**

1. Wählen Sie mindestens eine abhängige Variable aus.
2. Wählen Sie mindestens einen Faktor oder eine Kovariante aus.

Optional können Sie auf der Registerkarte "Variablen" die Methode zur Neuskalierung der Kovariaten ändern. Folgende Optionen stehen zur Auswahl:

- **Standardisiert.** Subtraktion des Mittelwerts und Division durch die Standardabweichung,  $(x - \text{Mittelwert})/s$ .
- **Normalisiert.** Subtraktion des Mittelwerts und Division durch den Bereich,  $(x - \text{Min})/(\text{Max} - \text{Min})$ . Normalisierte Werte liegen im Bereich zwischen 0 und 1.
- **Angepasst normalisiert.** Angepasste Version der Subtraktion des Mittelwerts und Division durch den Bereich,  $[2 * (x - \text{Min})/(\text{Max} - \text{Min})] - 1$ . Angepasste, normalisierte Werte liegen im Bereich zwischen -1 und 1.
- **Keine.** Keine Neuskalierung der Kovariaten.

Felder mit unbekanntem Messniveau

Der Messniveau-Alert wird angezeigt, wenn das Messniveau für mindestens eine Variable (ein Feld) im Dataset unbekannt ist. Da sich das Messniveau auf die Berechnung der Ergebnisse für diese Prozedur auswirkt, müssen alle Variablen ein definiertes Messniveau aufweisen.

**Daten durchsuchen.** Liest die Daten im aktiven Dataset und weist allen Feldern, deren Messniveau zurzeit nicht bekannt ist, das Standardmessniveau zu. Bei großen Datasets kann dieser Vorgang einige Zeit in Anspruch nehmen.

**Manuell zuweisen.** Öffnet ein Dialogfeld, in dem alle Felder mit unbekanntem Messniveau aufgeführt werden. Mit diesem Dialogfeld können Sie diesen Feldern ein Messniveau zuweisen. Außerdem können Sie in der Variablenansicht des Dateneditors ein Messniveau zuweisen.

Da das Messniveau für diese Prozedur bedeutsam ist, können Sie erst dann auf das Dialogfeld zur Ausführung dieser Prozedur zugreifen, wenn für alle Felder ein Messniveau definiert wurde.

## **Partitionen**

**Partitionsdataset.** Diese Gruppe gibt die Methode zur Partitionierung des aktiven Datasets in eine Trainings-, eine Test- und eine Holdout-Stichprobe an. Die **Trainingsstichprobe** umfasst die Datensätze, die zum Trainieren des neuronalen Netzes verwendet wurden. Ein gewisser Prozentsatz der Fälle im Dataset muss der Trainingsstichprobe zugewiesen werden, um ein Modell zu erhalten. Die **Teststichprobe** ist ein unabhängiges Set von Datensätzen, die verwendet werden, um den Fehler während des Trainings aufzuzeichnen und dadurch ein Übertrainieren zu vermeiden. Es wird dringend empfohlen, eine Teststichprobe zu erstellen. Das Netztraining ist in der Regel am effizientesten, wenn die Teststichprobe kleiner ist als die Trainingsstichprobe. Die **Holdout-Stichprobe** ist ein weiteres unabhängiges Set von Datensätzen, das zur Bewertung des endgültigen neuronalen Netzes verwendet wird. Der Fehler für die Holdout-Stichprobe

bietet eine "ehrliche" Schätzung der Vorhersagekraft des Modells, da die Holdout-Fälle (die Fälle in der Holdout-Stichprobe) nicht zur Erstellung des Modells verwendet wurden.

- **Fälle auf der Grundlage der relativen Anzahl an Fällen zufällig zuweisen.** Geben Sie die relative Anzahl (Verhältnis) der Fälle an, die den einzelnen Stichproben (Training, Test und Holdout) nach dem Zufallsprinzip zugewiesen werden sollen. Die Spalte % gibt auf der Grundlage der von Ihnen angegebenen Werte für die relative Anzahl den Prozentsatz der Fälle an, die den einzelnen Stichproben zugewiesen werden.

Die Angabe von 7, 3, 0 als relative Anzahl für Training-, Test- und Holdout-Stichprobe entspricht 70 %, 30 % und 0 %. Die Angabe von 2, 1, 1 als Werte für die relative Anzahl entspricht 50 %, 25 % und 25 %. 1, 1, 1 entspricht der Aufteilung des Datensets in drei gleich große Teile für Training, Test und Holdout.

- **Partitionsvariable zum Zuweisen von Fällen verwenden.** Geben Sie eine numerische Variable an, die jeden Fall im aktiven Dataset der Trainings-, Test oder Holdout-Stichprobe zuweist. Fälle mit einem positiven Wert für die Variable werden der Trainingsstichprobe zugewiesen, Fälle mit dem Wert 0 der Teststichprobe und Fälle mit einem negativen Wert der Holdout-Stichprobe. Fälle mit einem systemdefiniert fehlenden Wert werden aus der Analyse ausgeschlossen. Alle benutzerdefiniert fehlenden Werte für die Partitionsvariable werden immer als gültig behandelt.

*Hinweis:* Die Verwendung einer Partitionsvariablen garantiert keine identischen Ergebnisse bei späteren Ausführungen der Prozedur. Weitere Informationen finden Sie unter "Reproduzieren der Ergebnisse" im Hauptthema Mehrschichtiges Perzeptron.

## Architektur

Auf der Registerkarte "Architektur" können Sie die Struktur des Netzes angeben. Die Prozedur kann automatisch die "beste" Architektur auswählen, Sie können aber auch eine benutzerdefinierte Architektur angeben.

Mit der automatischen Architekturauswahl wird ein Netz mit genau einer verborgenen Schicht erstellt. Geben Sie die Mindest- und die Höchstzahl an Einheiten an, die in der verborgenen Schicht zulässig sein sollen. Die automatische Architekturauswahl berechnet daraus die "beste" Anzahl an Einheiten in der verborgenen Schicht. Die automatische Architekturauswahl verwendet die standardmäßigen Aktivierungsfunktionen für die verborgene Schichten und Ausgabeschichten.

Mit der benutzerdefinierten Architekturauswahl verfügen Sie über umfassende Kontrolle über die verborgenen Schichten und Ausgabeschichten. Dies ist insbesondere dann von Vorteil, wenn Sie im Voraus wissen, welche Architektur Sie wünschen, oder um eine Feinabstimmung der Ergebnisse der automatischen Architekturauswahl vorzunehmen.

### Verborgene Schichten

Die verborgene Schicht enthält nicht sichtbare Netzknoten (Einheiten). Jede verborgene Schicht ist eine Funktion der gewichteten Summe der Eingaben. Bei der Funktion handelt es sich um die Aktivierungsfunktion und die Werte der Gewichtungen richten sich nach dem Schätzalgorithmus. Wenn das Netz eine zweite verborgene Schicht enthält, ist jede verborgene Einheit in der zweiten Schicht eine Funktion der gewichteten Summe der Einheiten in der ersten verborgenen Schicht. In beiden Schichten wird dieselbe Aktivierungsfunktion verwendet.

*Anzahl der verborgenen Schichten.* Ein Mehrschicht-Perzeptron kann eine oder zwei verborgene Schichten enthalten.

*Aktivierungsfunktion.* Die Aktivierungsfunktion "verknüpft" die gewichteten Summen der Einheiten in einer Schicht mit den Werten der Einheiten in der nachfolgenden Schicht.

- **Hyperbeltangens.** Diese Funktion hat das folgende Format:  $\gamma(c) = \tanh(c) = (e^c - e^{-c}) / (e^c + e^{-c})$ . Sie verwendet Argumente mit reellen Werten und transformiert sie in den Bereich (-1, 1). Bei Verwendung der automatischen Architekturauswahl wird diese Aktivierungsfunktion für alle Einheiten in den verborgenen Schichten verwendet.
- **Sigmoid.** Diese Funktion hat das folgende Format:  $\gamma(c) = 1 / (1 + e^{-c})$ . Sie verwendet Argumente mit reellen Werten und transformiert sie in den Bereich (0, 1).

*Anzahl der Einheiten.* Die Anzahl der Einheiten in den einzelnen verborgenen Schichten kann explizit festgelegt oder automatisch durch den Schätzalgorithmus bestimmt werden.

Ausgabeschicht

Die Ausgabeschicht enthält die Zielvariablen (abhängigen Variablen).

*Aktivierungsfunktion.* Die Aktivierungsfunktion "verknüpft" die gewichteten Summen der Einheiten in einer Schicht mit den Werten der Einheiten in der nachfolgenden Schicht.

- **Identität.** Diese Funktion hat das Format:  $\gamma(c) = c$ . Sie nimmt reale Argumente an und gibt sie unverändert zurück. Bei Verwendung der automatischen Architekturauswahl wird diese Aktivierungsfunktion für Einheiten in der Ausgabeschicht verwendet, sofern metrische abhängige Variablen vorliegen.
- **Softmax.** Diese Funktion hat das folgende Format:  $\gamma(c_k) = \exp(c_k) / \sum_j \exp(c_j)$ . Sie nimmt einen Vektor mit realen Argumenten und transformiert ihn in einen Vektor, dessen Elemente in den Bereich (0, 1) fallen und sich auf 1 summieren. Softmax ist nur verfügbar, wenn alle abhängigen Variablen kategorial sind. Bei Verwendung der automatischen Architekturauswahl wird diese Aktivierungsfunktion für Einheiten in der Ausgabeschicht verwendet, sofern alle abhängigen Variablen kategorial sind.
- **Hyperbeltangens.** Diese Funktion hat das folgende Format:  $\gamma(c) = \tanh(c) = (e^{-c} - e^c) / (e^{-c} + e^c)$ . Sie verwendet Argumente mit reellen Werten und transformiert sie in den Bereich (-1, 1).
- **Sigmoid.** Diese Funktion hat das folgende Format:  $\gamma(c) = 1 / (1 + e^{-c})$ . Sie verwendet Argumente mit reellen Werten und transformiert sie in den Bereich (0, 1).

**Neuskalierung der metrischen abhängigen Variablen.** Diese Steuerelemente sind nur verfügbar, wenn mindestens eine metrische abhängige Variable ausgewählt wurde.

- **Standardisiert.** Subtraktion des Mittelwerts und Division durch die Standardabweichung,  $(x - \text{Mittelwert}) / s$ .
- **Normalisiert.** Subtraktion des Mittelwerts und Division durch den Bereich,  $(x - \text{Min}) / (\text{Max} - \text{Min})$ . Normalisierte Werte liegen im Bereich zwischen 0 und 1. Dies ist die erforderliche Rescaling-Methode für skalenabhängige Variablen, wenn die Ausgabeschicht die Sigmoid-Aktivierungsfunktion verwendet. Die Korrekturoption gibt einen kleinen  $\epsilon$ -Wert an, der als Korrektur der Neuskalierungsformel verwendet wird. Durch diese Korrektur wird sichergestellt, dass alle neu skalierten Werte abhängiger Variablen innerhalb des Bereichs der Aktivierungsfunktion liegen. Insbesondere definieren die Werte 0 und 1, die in der unkorrigierten Formel vorkommen, wenn  $x$  den Mindest- oder Maximalwert annimmt, zwar die Grenzen des Bereichs der Sigmoid-Funktion, liegen jedoch nicht innerhalb dieses Bereichs. Die korrigierte Formel lautet  $[x - (\text{Min} - \epsilon)] / [(\text{Max} + \epsilon) - (\text{Min} - \epsilon)]$ . Geben Sie eine Zahl größer-gleich 0 ein.
- **Angepasst normalisiert.** Angepasste Version der Subtraktion des Mittelwerts und Division durch den Bereich,  $[2 * (x - \text{Min}) / (\text{Max} - \text{Min})] - 1$ . Angepasste normalisierte Werte liegen zwischen -1 und 1. Dies ist die erforderliche Reskalierungsmethode für skalenabhängige Variablen, wenn die Ausgabeschicht die Funktion zur Aktivierung des Hyperboli-Tangens verwendet. Die Korrekturoption gibt einen kleinen  $\epsilon$ -Wert an, der als Korrektur der Neuskalierungsformel verwendet wird. Durch diese Korrektur wird sichergestellt, dass alle neu skalierten Werte abhängiger Variablen innerhalb des Bereichs der Aktivierungsfunktion liegen. Insbesondere definieren die Werte -1 und 1, die in der unkorrigierten Formel vorkommen, wenn  $x$  den Mindest- oder Maximalwert annimmt, zwar die Grenzen des Bereichs der Hyperbeltangens-Funktion, liegen jedoch nicht innerhalb dieses Bereichs. Die korrigierte Formel lautet  $\{2 * [(x - (\text{Min} - \epsilon))] / [(\text{Max} + \epsilon) - (\text{Min} - \epsilon)]\} - 1$ . Geben Sie eine Zahl größer-gleich 0 ein.
- **Keine.** Keine Neuskalierung metrischer abhängiger Variablen.

## Training

Auf der Registerkarte "Training" können Sie angeben, wie das Netz trainiert werden sollte. Die Art des Trainings und der Optimierungsalgorithmus bestimmen, welche Trainingsoptionen verfügbar sind.

**Art des Trainings.** Die Art des Trainings bestimmt, wie das Netz die Datensätze verarbeitet. Wählen Sie eine der folgenden Trainingsarten:

- **Stapel.** Aktualisiert die synaptischen Gewichtungen erst nach dem Durchlauf sämtlicher Trainingsdatensätze. Beim Batch-Training werden also die Daten aus allen Datensätzen im Trainingsdataset verwendet. Batch-Training wird häufig bevorzugt, da damit der Gesamtfehler unmittelbar minimiert wird.

Allerdings kann beim Batch-Training eine sehr häufige Aktualisierung der Gewichtungen erforderlich sein, bis eine der Stoppregeln erfüllt ist, sodass sehr viele Datendurchläufe notwendig sein können. Es eignet sich vor allem für "kleinere" Datensets.

- **Online.** Aktualisiert die synaptischen Gewichtungen nach jedem einzelnen Trainingsdatensatz. Beim Online-Training werden also jeweils immer nur die Daten aus einem einzigen Datensatz verwendet. Das Online-Training ruft ständig einen Datensatz ab und aktualisiert die Gewichtungen, bis eine der Stoppregeln erfüllt ist. Wenn alle Datensätze einmal verwendet wurden und keine der Stoppregeln erfüllt ist, wird der Prozess mit einem erneuten Durchlauf der Datensätze fortgesetzt. Online-Schulungen sind der Stapelverarbeitung für "größere" Datensets mit zugeordneten Prädiktoren überlegen, d. h. wenn viele Datensätze und viele Eingaben vorhanden sind und ihre Werte nicht voneinander unabhängig sind, kann die Online-Schulung schneller eine angemessene Antwort erhalten als das Batch-Training.
- **Mini-Batch.** Unterteilt die Trainingsdatensätze in ungefähr gleich große Gruppen und aktualisiert dann die synaptischen Gewichtungen jeweils nach dem Durchlauf einer Gruppe. Beim Mini-Batch-Training werden also Informationen aus einer Gruppe von Datensätzen verwendet. Anschließend wird die Datengruppe, falls erforderlich, erneut verwendet. Mini-Batch-Training stellt einen Kompromiss zwischen Batch-Training und Online-Training dar und eignet sich am besten für "mittelgroße" Datensets. Die Prozedur kann die Anzahl der Trainingsdatensätze pro Mini-Batch automatisch festlegen. Sie können jedoch auch eine ganze Zahl größer 1 und kleiner-gleich der maximalen Anzahl der im Arbeitsspeicher zu speichernden Fälle angeben. Sie können die maximale Anzahl der im Arbeitsspeicher zu speichernden Fälle auf der Registerkarte [Optionen](#) festlegen.

**Optimierungsalgorithmus.** Diese Methode wird zur Schätzung der synaptischen Gewichtungen verwendet.

- **Skalierter konjugierter Gradient.** Die Annahmen, die die Verwendung von konjugierten Gradientenmethoden rechtfertigen, gelten nur für Batch-Training-Typen, so dass diese Methode nicht für Online- oder Mini-Batch-Schulungen verfügbar ist.
- **Gradientenabstieg.** Diese Methode muss mit Online- oder Mini-Batch-Training verwendet werden; sie kann auch mit Batch-Training verwendet werden.

**Trainingsoptionen.** Die Trainingsoptionen ermöglichen eine Feinabstimmung des Optimierungsalgorithmus. Im Allgemeinen müssen Sie diese Einstellungen nur ändern, wenn beim Netz Probleme mit der Schätzung auftreten.

Folgende Trainingsoptionen stehen für den Algorithmus mit skaliertem konjugiertem Gradienten zur Verfügung:

- **Anfangs-Lambda.** Der ursprüngliche Wert des Lambda-Parameters für den Algorithmus mit skaliertem konjugiertem Gradienten. Geben Sie einen Wert größer als 0 und kleiner als 0,000001 ein.
- **Anfangs-Sigma.** Der ursprüngliche Wert des Sigma-Parameters für den Algorithmus mit skaliertem konjugiertem Gradienten. Geben Sie einen Wert größer als 0 und kleiner als 0,0001 ein.
- **Intervallzentrum und Intervalloffset.** Intervallzentrum ( $a_0$ ) und Intervalloffset ( $a$ ) definieren das Intervall  $[a_0 - a, a_0 + a]$ , in dem bei Verwendung der simulierten Abkühlung Gewichtungsvektoren nach dem Zufallsprinzip erstellt werden. Die simulierte Abkühlung wird verwendet, um während der Anwendung des Optimierungsalgorithmus aus einem lokalen Minimum ausbrechen zu können, um das globale Minimum zu finden. Dieser Ansatz wird bei der Gewichtungsinitalisierung und bei der automatischen Architekturauswahl verwendet. Geben Sie eine Zahl für das Intervallzentrum und eine Zahl über 0 für den Intervallversatz an.

Folgende Trainingsoptionen stehen für den Gradientenabstiegsalgorithmus zur Verfügung:

- **Anfängliche Lernrate.** Der ursprüngliche Wert der Lernrate für den Gradientenabstiegsalgorithmus. Bei einer höheren Lernrate erfolgt das Training des Netzes schneller, kann jedoch möglicherweise instabil werden. Geben Sie einen Wert größer 0 an.
- **Untergrenze der Lernrate.** Die Untergrenze der Lernrate für den Gradientenabstiegsalgorithmus. Diese Einstellung gilt nur für Online-Training und Mini-Batch-Training. Geben Sie einen Wert ein, der größer als 0 und kleiner als die anfängliche Lernrate ist.

- **Momentum.** Der ursprüngliche Momentum-Parameter für den Gradientenabstiegsalgorithmus. Der Momentum-Term (Impulsterm) trägt zur Vermeidung von Instabilitäten bei, die durch eine zu hohe Lernrate verursacht werden. Geben Sie einen Wert größer 0 an.
- **Lernratenreduzierung, in Epochen.** Die Anzahl der Epochen ( $p$ ) oder Datendurchläufe der Trainingsstichprobe, die zur Reduzierung der anfänglichen Lernrate auf die Untergrenze der Lernrate erforderlich sind, wenn beim Online- oder Mini-Batch-Training Gradientenabstieg verwendet wird. Dadurch können Sie den Faktor für den Lernratenverfall  $\beta = (1/p K) * \ln(\eta_0/\eta_{\text{niedrig}})$  steuern. Dabei ist  $\eta_0$  die anfängliche Lernrate,  $\eta_{\text{niedrig}}$  ist die Untergrenze der Lernrate und  $K$  ist die Gesamtzahl der Mini-Batches (bzw. beim Online-Training die Anzahl der Trainingsdatensätze) im Trainingsdataset. Geben Sie eine ganze Zahl größer 0 an.

## Ausgabe

**Netzstruktur.** Zeigt zusammenfassende Informationen über das neuronale Netz an.

- **Beschreibung.** Zeigt Informationen über das neuronale Netz an, einschließlich der abhängigen Variablen, der Anzahl der Ein- und Ausgabeneinheiten, der Anzahl der verdeckten Ebenen und Einheiten und der Aktivierungsfunktionen.
- **Diagramm.** Zeigt das Netzdiagramm als nicht bearbeitbares Diagramm an. Beachten Sie: Mit steigender Anzahl an Kovariaten und Faktorstufen wird das Diagramm schwerer zu interpretieren.
- **Synaptische Gewichtungen.** Zeigt die Koeffizientenschätzungen an, die die Beziehung zwischen den Einheiten in einer bestimmten Schicht und den Einheiten in der nächsten Schicht anzeigen. Die synaptischen Gewichtungen beruhen auf der Trainingsstichprobe, selbst wenn das aktive Dataset in Trainings-, Test- und Holdout-Daten partitioniert ist. Beachten Sie, dass die Anzahl der synaptischen Gewichtungen recht groß werden kann und dass diese Gewichtungen im Allgemeinen nicht zur Interpretation der Netzergebnisse verwendet werden.

**Netzleistung.** Zeigt die Ergebnisse an, die verwendet werden, um zu bestimmen, ob das Modell "gut" ist. *Hinweis:* Die Diagramme in dieser Gruppe beruhen auf der Kombination aus Trainings- und Teststichprobe oder nur auf der Trainingsstichprobe, wenn keine Teststichprobe vorhanden ist.

- **Modellzusammenfassung.** Zeigt eine Zusammenfassung der Ergebnisse des neuronalen Netzes nach Partition und insgesamt an, einschließlich der folgenden Werte: Fehler, Relativer Fehler oder Prozentsatz der falschen Vorhersagen, zum Beenden des Trainings verwendete Stopregel und Trainingszeit.

Bei Anwendung der Aktivierungsfunktion "Identität", "Sigmoid" oder "Hyperbeltangens" auf die Ausgabeschicht handelt es sich um den Quadratsummenfehler. Bei Anwendung der Aktivierungsfunktion "Softmax" auf die Ausgabeschicht handelt es sich um den Kreuzentropiefehler.

Die relativen Fehler oder Prozentsätze der falschen Vorhersagen werden in Abhängigkeit von den Messniveaus der abhängigen Variablen angezeigt. Wenn eine abhängige Variable ein metrisches Messniveau aufweist, wird der durchschnittliche relative Gesamtfehler (relativ zum Mittelwertmodell) angezeigt. Wenn alle abhängigen Variablen kategorial sind, wird der durchschnittliche Prozentsatz der falschen Vorhersagen angezeigt. Die relativen Fehler oder Prozentsätze der falschen Vorhersagen werden jeweils für die einzelnen abhängigen Variablen angezeigt.

- **Klassifikationsergebnisse.** Zeigt eine Klassifikationstabelle für die einzelnen kategorialen abhängigen Variablen (nach Partition und insgesamt) an. Jede Tabelle gibt für jede Kategorie abhängiger Variablen die Anzahl der korrekt und nicht korrekt klassifizierten Fälle an. Der Prozentsatz der Gesamtzahl der Fälle, die korrekt klassifiziert wurden, wird ebenfalls angegeben.
- **ROC-Kurve.** Zeigt eine ROC-Kurve (Receiver Operating Characteristic) für jede kategoriale abhängige Variable an. Außerdem wird eine Tabelle angezeigt, die die Fläche unter den einzelnen Kurven angibt. Bei jeder abhängigen Variablen zeigt das ROC-Diagramm jeweils genau eine Kurve für jede Kategorie an. Wenn die abhängige Variable zwei Kategorien aufweist, behandelt jede Kurve die fragliche Kategorie als positiven Zustand gegenüber der anderen Kategorie. Wenn die abhängige Variable mehr als zwei Kategorien aufweist, behandelt jede Kurve die fragliche Kategorie als positiven Zustand gegenüber allen anderen Kategorien.

- **Kumulative Gewinndiagramme.** Zeigt für jede kategoriale abhängige Variable ein kumulatives Gewinndiagramm an. Die Anzeige einer Kurve für jede Kategorie der abhängigen Variablen verhält sich wie bei ROC-Kurven.
- **Liftdiagramm.** Zeigt für jede kategoriale abhängige Variable ein Liftdiagramm an. Die Anzeige einer Kurve für jede Kategorie der abhängigen Variablen verhält sich wie bei ROC-Kurven.
- **Diagramm "Vorhergesagt/Beobachtet".** Zeigt für jede abhängige Variable ein Diagramm an, das die vorhergesagten Werte in Abhängigkeit von den beobachteten Werten angibt. Bei kategorialen abhängigen Variablen werden für jede Antwortkategorie gruppierte Boxplots der vorhergesagten Pseudowahrscheinlichkeiten angezeigt, wobei die Kategorie der beobachteten Antworten als Clustervariable fungiert. Bei metrischen abhängigen Variablen wird ein Streudiagramm angezeigt.
- **Diagramm "Residuum/Vorhergesagt".** Zeigt für jede metrische abhängige Variable ein Diagramm an, das die Residuen in Abhängigkeit von den vorhergesagten Werten angibt. Es sollte kein Muster zwischen Residuen und vorhergesagten Werten zu beobachten sein. Dieses Diagramm wird nur bei metrischen abhängigen Variablen erstellt.

**Zusammenfassung der Fallverarbeitung.** Zeigt die Tabelle mit der Zusammenfassung der Fallverarbeitung an, die die Anzahl der in der Analyse ein- und ausgeschlossenen Fälle zusammenfasst (insgesamt und nach Trainings-, Test- und Holdout-Stichprobe geordnet).

**Wichtigkeitsanalyse für unabhängige Variablen.** Führt eine Sensitivitätsanalyse durch, mit der die Wichtigkeit der einzelnen Prädiktoren für die Bestimmung des neuronalen Netzes berechnet wird. Die Analyse beruht auf der Kombination aus Trainings- und Teststichprobe oder nur auf der Trainingsstichprobe, wenn keine Teststichprobe vorhanden ist. Dadurch werden eine Tabelle und ein Diagramm erstellt, die die Wichtigkeit und die normalisierte Wichtigkeit für die einzelnen Prädiktoren anzeigen. Beachten Sie, dass die Sensitivitätsanalyse rechenintensiv und zeitaufwendig ist, wenn eine große Anzahl an Prädiktoren oder Fällen vorliegt.

## Speichern

Auf der Registerkarte "Speichern" können Vorhersagen im Dataset als Variablen gespeichert werden.

- **Für jede abhängige Variable vorhergesagten Wert oder Kategorie speichern.** Damit wird bei metrischen abhängigen Variablen der vorhergesagte Wert und bei kategorialen abhängigen Variablen die vorhergesagte Kategorie gespeichert.
- **Für jede abhängige Variable vorhergesagte Pseudowahrscheinlichkeit oder Kategorie speichern** Damit werden bei kategorialen abhängigen Variablen die vorhergesagten Pseudowahrscheinlichkeiten gespeichert. Für die ersten  $n$  Kategorien wird eine separate Variable gespeichert. Dabei wird  $n$  in der Spalte **Zu speichernde Kategorien** angegeben.

**Namen der gespeicherten Variablen.** Durch eine automatische Generierung von Namen wird sichergestellt, dass Ihre Arbeit nicht verloren geht. Mit benutzerdefinierten Namen können Sie Ergebnisse aus früheren Durchgängen verwerfen/ersetzen, ohne zuerst die gespeicherten Variablen im Dateneditor löschen zu müssen.

Wahrscheinlichkeiten und Pseudowahrscheinlichkeiten

Kategoriale abhängige Variablen mit Softmax-Aktivierung und Kreuzentropiefehler weisen einen vorhergesagten Wert für jede Kategorie auf, wobei die einzelnen vorhergesagten Werte jeweils die Wahrscheinlichkeit angeben, dass der Fall zu der betreffenden Kategorie gehört.

Kategoriale abhängige Variablen mit Quadratsummenfehler weisen einen vorhergesagten Wert für jede Kategorie auf, die vorhergesagten Werte können jedoch nicht als Wahrscheinlichkeiten interpretiert werden. Die Prozedur speichert diese vorhergesagten Pseudowahrscheinlichkeiten, selbst wenn sie kleiner als 0 oder größer als 1 sind oder wenn die Summe für eine abhängige Variable nicht 1 ergibt.

ROC-Diagramme, kumulative Gewinndiagramme und Liftdiagramme (siehe „Ausgabe“ auf Seite 9) werden auf der Basis von Pseudowahrscheinlichkeiten erstellt. Für den Fall, dass eine der Pseudo-Wahrscheinlichkeiten kleiner als 0 oder größer als 1 ist oder die Summe für eine gegebene Variable nicht 1 ist, werden sie zunächst auf einen Wert zwischen 0 und 1 und auf die Summe 1 skaliert. Pseudowahrscheinlichkeiten werden durch Division durch ihre Summe skaliert. Wenn ein Fall beispielsweise für eine

abhängige Variable mit drei Kategorien vorhergesagte Pseudowahrscheinlichkeiten von 0,50, 0,60 und 0,40 aufweist, wird jede Pseudowahrscheinlichkeit durch die Summe 1,50 dividiert, woraus sich die Werte 0,33, 0,40 und 0,27 ergeben.

Wenn negative Pseudowahrscheinlichkeiten vorliegen, werden vor der oben beschriebenen Neuskalierung alle Pseudowahrscheinlichkeiten jeweils mit dem Betrag der niedrigsten Wahrscheinlichkeit addiert. Wenn die Pseudowahrscheinlichkeiten beispielsweise -0,30, 0,50 und 1,30 betragen, müssen Sie zunächst 0,30 zu jedem Wert addieren und erhalten somit die Werte 0,00, 0,80 und 1,60. Als Nächstes dividieren Sie die einzelnen neuen Werte durch die Summe 2,40, wodurch sich die Werte 0,00, 0,33 und 0,67 ergeben.

## Export

Die Registerkarte "Export" dient zum Speichern der Schätzungen der synaptischen Gewichtungen für die einzelnen abhängigen Variablen in einer XML-Datei (PMML-Datei). Anhand dieser Modelldatei können Sie die Modellinformationen zu Scoring-Zwecken auf andere Datendateien anwenden. Diese Option ist nicht verfügbar, wenn aufgeteilte Dateien definiert wurden.

## Optionen

**Benutzerdefiniert fehlende Werte.** Faktoren müssen gültige Werte für einen Fall aufweisen, um in die Analyse aufgenommen zu werden. Mit diesen Steuerelementen legen Sie fest, ob benutzerdefiniert fehlende Werte bei den Faktoren und kategorialen abhängigen Variablen als gültige Werte behandelt werden sollen.

**Stoppregeln.** Dies sind die Regeln, die festlegen, wann das Training des neuronalen Netzes abgebrochen werden soll. Das Training erfolgt über mindestens einen Datendurchlauf. Anschließend kann das Training gemäß den folgenden Kriterien beendet werden, die in der angegebenen Reihenfolge überprüft werden. In den folgenden Definitionen für Stoppregeln entspricht ein Schritt bei den Methoden "Online" und "Mini-Batch" einem Datendurchlauf, bei der Batchmethode einer Iteration.

- **Maximale Anzahl an Schritten ohne Verringerung des Fehlers.** Die Anzahl der Schritte, die zulässig sind, bevor eine Prüfung auf Verringerung des Fehlers erfolgt. Wenn nach der angegebenen Anzahl an Schritten keine Verringerung des Fehlers zu verzeichnen ist, wird das Training beendet. Geben Sie eine ganze Zahl größer 0 an. Sie können auch angeben, welches Datenmuster verwendet wird, um den Fehler zu berechnen. Bei **Automatisch auswählen** wird die Teststichprobe verwendet, sofern vorhanden. Andernfalls wird die Trainingsstichprobe verwendet. Beachten Sie, dass beim Batch-Training der Fehler bei der Trainingsstichprobe garantiert nach jedem Datendurchlauf kleiner wird, daher kann diese Option nur auf das Batch-Training angewendet werden, wenn eine Teststichprobe vorhanden ist. Mit **Trainings- und Testdaten** wird der Fehler für jede dieser Stichproben geprüft. Diese Option gilt nur, wenn eine Teststichprobe vorhanden ist.

*Hinweis:* Nach jedem vollständigen Datendurchlauf ist beim Online- und Mini-Batch-Training ein zusätzlicher Datendurchlauf zur Berechnung des Trainingsfehlers erforderlich. Dieser zusätzliche Datendurchlauf kann das Training erheblich verlangsamen. Daher wird allgemein empfohlen, in jedem Fall eine Teststichprobe anzugeben und **Automatisch auswählen** zu verwenden.

- **Maximale Trainingszeit.** Sie können wählen, ob sie eine maximale Minutenanzahl für die Ausführung des Algorithmus angeben wollen. Geben Sie einen Wert größer 0 an.
- **Maximale Anzahl an Trainingsepochen.** Die maximal zulässige Anzahl an Epochen (Datendurchläufen). Wenn die maximale Anzahl an Epochen überschritten ist, wird das Training beendet. Geben Sie eine ganze Zahl größer 0 an.
- **Minimale relative Änderung beim Trainingsfehler.** Das Training wird beendet, wenn die relative Änderung beim Trainingsfehler im Vergleich zum vorherigen Schritt kleiner ist als der Kriterienwert. Geben Sie einen Wert größer 0 an. Bei Online- und Mini-Batch-Training wird dieses Kriterium ignoriert, wenn nur Testdaten verwendet werden, um den Fehler zu berechnen.
- **Minimale relative Änderung beim Trainingsfehlerquotienten.** Das Training wird beendet, wenn der Quotient aus Trainingsfehler und Fehler des Nullmodells kleiner ist als der Kriterienwert. Das Nullmodell sagt den Durchschnittswert für alle abhängigen Variablen voraus. Geben Sie einen Wert größer 0

an. Bei Online- und Mini-Batch-Training wird dieses Kriterium ignoriert, wenn nur Testdaten verwendet werden, um den Fehler zu berechnen.

**Maximale Anzahl der im Arbeitsspeicher zu speichernden Fälle.** Dadurch werden folgende Einstellungen innerhalb der Algorithmen mit mehrschichtigem Perzeptron gesteuert. Geben Sie eine ganze Zahl größer als 1 an.

- Bei der automatischen Architekturauswahl beträgt die zur Bestimmung der Netzarchitektur verwendete Stichprobe  $\min(1000, memsize)$ , wobei  $memsize$  die maximale Anzahl der im Arbeitsspeicher zu speichernden Fälle ist.
- Beim Mini-Batch-Training mit automatischer Berechnung der Anzahl an Mini-Batches, beträgt die Anzahl der Mini-Batches  $\min(\max(M/10, 2), memsize)$ , wobei  $M$  die Anzahl der Fälle in der Trainingsstichprobe ist.

## Radiale Basisfunktion

Die Prozedur "Radiale Basisfunktion" (RBF) erstellt ein Vorhersagemodell für mindestens eine abhängige Variable (Zielvariable), das auf den Werten der Prädiktorvariablen beruht.

**Beispiel.** Ein Telekommunikationsanbieter hat seinen Kundenstamm nach Serviceverwendungsmustern eingeteilt und die Kunden in vier Gruppen unterteilt. Ein RBF-Netz, das demografische Daten zur Vorhersage der Gruppenzugehörigkeit verwendet, ermöglicht es dem Unternehmen, Angebote für einzelne Interessenten anzupassen.












Erläuterung der Daten

**Abhängige Variablen.** Die abhängigen Variablen können wie folgt gestaltet sein:

- *Nominal.* Eine Variable kann als nominal behandelt werden, wenn ihre Werte Kategorien ohne interne Rangfolge darstellen (z. B. die Abteilung des Unternehmens, in der ein Mitarbeiter arbeitet). Beispiele für nominale Variablen sind Region, Postleitzahl oder Religionszugehörigkeit.
- *Ordinal.* Eine Variable kann als ordinal behandelt werden, wenn ihre Werte Kategorien mit einer intrinsischen Rangfolge darstellen (z. B. Grad der Servicezufriedenheit von hoch unzufrieden bis hoch zufrieden). Ordinale Variablen treten beispielsweise bei Einstellungsmessungen (Zufriedenheit oder Vertrauen) und bei Präferenzbeurteilungen auf.
- *Skala.* Eine Variable kann als metrisch (stetig) behandelt werden, wenn ihre Werte geordnete Kategorien mit einer aussagekräftigen Metrik darstellen, sodass Abstandsvergleiche zwischen Werten angemessen sind. Metrische Variablen sind beispielsweise Alter (in Jahren) oder Einkommen (in Geldeinheiten).

Bei der Prozedur wird davon ausgegangen, dass allen abhängigen Variablen das richtige Messniveau zugewiesen wurde. Sie können das Messniveau für eine Variable jedoch vorübergehend ändern. Klicken Sie hierzu mit der rechten Maustaste auf die Variable in der Liste der Quellenvariablen und wählen Sie das gewünschte Messniveau im Pop-up-Menü aus.

Messniveau und Datentyp sind durch ein Symbol neben der jeweiligen Variablen in der Variablenliste gekennzeichnet:

	Numerisch	Zeichenfolge	Datum	Zeit
Metrisch (stetig)		nicht zutreffend		
Ordinalzahl				
Nominal				



**Prädiktorvariablen.** Prädiktoren können als Faktoren (kategorial) oder als Kovariaten (metrisch) angegeben werden.

**Codierung für kategoriale Variablen.** Die Prozedur codiert vorübergehend für die Dauer ihrer Ausführung kategoriale Prädiktoren und abhängige Variablen mithilfe der "Eins-aus-c"-Codierung neu. Wenn es c-Kategorien einer Variablen gibt, wird die Variable als C-Vektoren gespeichert, wobei die erste Kategorie (1, 0, ..., 0), die nächste Kategorie (0, 1, 0, ..., 0), ..., und die letzte Kategorie (0, 0, ..., 0, 1).

Dieses Codierungsschema erhöht die Anzahl der synaptischen Gewichtungen und kann zu einer Verlangsamung des Trainings führen, "kompaktere" Codierungsmethoden führen jedoch in der Regel zu neuronalen Netzen mit geringer Anpassungsgüte. Wenn Ihr Netztraining sehr langsam voranschreitet, versuchen Sie, die Anzahl der Kategorien in Ihren kategorialen Prädiktoren zu reduzieren, indem Sie ähnliche Kategorien kombinieren oder Fälle mit extrem seltenen Kategorien löschen.

Jede "Eins-aus-c"-Codierung beruht auf den Trainingsdaten, selbst wenn eine Test- oder Holdout-Stichprobe definiert wurde (siehe „Partitionen“ auf Seite 14). Wenn also die Test- oder Holdout-Stichproben Fälle mit Prädiktorkategorien enthalten, die in den Trainingsdaten nicht vorhanden sind, werden diese Fälle nicht in der Prozedur oder beim Scoring verwendet. Wenn die Test- oder Holdout-Stichproben Fälle mit Kategorien abhängiger Variablen enthalten, die in den Trainingsdaten nicht vorhanden sind, werden diese Fälle zwar nicht in der Prozedur, jedoch möglicherweise beim Scoring verwendet.

**Neuskalierung.** Metrische abhängige Variablen und Kovariaten werden standardmäßig neu skaliert, um das Training des Netzes zu verbessern. Jede Neuskalierung beruht auf den Trainingsdaten, selbst wenn eine Test- oder Holdout-Stichprobe definiert wurde (siehe „Partitionen“ auf Seite 14). Das bedeutet, dass je nach Neuskalierungstyp Mittelwert, Standardabweichung, Mindestwert oder Maximalwert einer Kovariaten oder abhängigen Variablen ausschließlich anhand der Trainingsdaten berechnet werden. Wenn Sie eine Variable zur Festlegung von Partitionen angeben, müssen diese Kovariaten oder abhängigen Variablen in der Trainings-, Test- und Holdout-Stichprobe ähnliche Verteilungen aufweisen.

**Häufigkeitsgewichtungen.** Häufigkeitsgewichtungen werden von dieser Prozedur ignoriert.

**Reproduzieren der Ergebnisse.** Wenn Sie Ihre Ergebnisse exakt reproduzieren möchten, müssen Sie nicht nur dieselben Einstellungen für die Prozedur, sondern auch denselben Initialisierungswert für den Zufallszahlengenerator und dieselbe Datenreihenfolge verwenden. Weitere Details zu diesem Problem folgen:

- **Generierung von Zufallszahlen.** Die Prozedur verwendet Zufallszahlengenerierung während der Zufallszuweisung von Partitionen. Um zu einem späteren Zeitpunkt dieselben randomisierten Ergebnisse zu reproduzieren, müssen Sie vor jeder Ausführung der Prozedur "Radiale Basisfunktion" denselben Initialisierungswert für den Zufallszahlengenerator verwenden.
- **Fallreihenfolge.** Außerdem hängen die Ergebnisse von der Datenreihenfolge ab, da der Two-Step-Clusteralgorithmus zur Ermittlung der radialen Basisfunktionen verwendet wird.

Um die Auswirkungen der Reihenfolge zu minimieren, mischen Sie die Fälle in zufälliger Reihenfolge. Prüfen Sie daher die Stabilität einer bestimmten Lösung, indem Sie verschiedene Lösungen abrufen, bei denen die Fälle in einer unterschiedlichen, zufällig ausgewählten Reihenfolge sortiert sind. In Situationen mit extrem umfangreichen Dateien können mehrere Durchgänge mit jeweils einer Stichprobe von Fällen durchgeführt werden, die in unterschiedlicher, zufällig ausgewählter Reihenfolge sortiert ist.

Erstellen eines Netzes mit radialen Basisfunktionen

Wählen Sie in den Menüs Folgendes aus:

**Analysieren > Neuronale Netze > Radiale Basisfunktion...**

1. Wählen Sie mindestens eine abhängige Variable aus.
2. Wählen Sie mindestens einen Faktor oder eine Kovariante aus.

Optional können Sie auf der Registerkarte "Variablen" die Methode zur Neuskalierung der Kovariaten ändern. Gültige Auswahlmöglichkeiten sind:

- **Standardisiert.** Subtraktion des Mittelwerts und Division durch die Standardabweichung,  $(x - \text{Mittelwert})/s$ .

- **Normalisiert.** Subtraktion des Mittelwerts und Division durch den Bereich,  $(x-\text{Min})/(\text{Max}-\text{Min})$ . Normalisierte Werte liegen im Bereich zwischen 0 und 1.
- **Angepasst normalisiert.** Angepasste Version der Subtraktion des Mittelwerts und Division durch den Bereich,  $[2*(x-\text{Min})/(\text{Max}-\text{Min})]-1$ . Angepasste, normalisierte Werte liegen im Bereich zwischen -1 und 1.
- **Keine.** Keine Neuskalierung der Kovariaten.

Felder mit unbekanntem Messniveau

Der Messniveau-Alert wird angezeigt, wenn das Messniveau für mindestens eine Variable (ein Feld) im Dataset unbekannt ist. Da sich das Messniveau auf die Berechnung der Ergebnisse für diese Prozedur auswirkt, müssen alle Variablen ein definiertes Messniveau aufweisen.

**Daten durchsuchen.** Liest die Daten im aktiven Dataset und weist allen Feldern, deren Messniveau zurzeit nicht bekannt ist, das Standardmessniveau zu. Bei großen Datasets kann dieser Vorgang einige Zeit in Anspruch nehmen.

**Manuell zuweisen.** Öffnet ein Dialogfeld, in dem alle Felder mit unbekanntem Messniveau aufgeführt werden. Mit diesem Dialogfeld können Sie diesen Feldern ein Messniveau zuweisen. Außerdem können Sie in der Variablenansicht des Dateneditors ein Messniveau zuweisen.

Da das Messniveau für diese Prozedur bedeutsam ist, können Sie erst dann auf das Dialogfeld zur Ausführung dieser Prozedur zugreifen, wenn für alle Felder ein Messniveau definiert wurde.

## Partitionen

**Partitionsdataset.** Diese Gruppe gibt die Methode zur Partitionierung des aktiven Datasets in eine Trainings-, eine Test- und eine Holdout-Stichprobe an. Die **Trainingsstichprobe** umfasst die Datensätze, die zum Trainieren des neuronalen Netzes verwendet wurden. Ein gewisser Prozentsatz der Fälle im Dataset muss der Trainingsstichprobe zugewiesen werden, um ein Modell zu erhalten. Die **Teststichprobe** ist ein unabhängiges Set von Datensätzen, die verwendet werden, um den Fehler während des Trainings aufzuzeichnen und dadurch ein Übertrainieren zu vermeiden. Es wird dringend empfohlen, eine Teststichprobe zu erstellen. Das Netztraining ist in der Regel am effizientesten, wenn die Teststichprobe kleiner ist als die Trainingsstichprobe. Die **Holdout-Stichprobe** ist ein weiteres unabhängiges Set von Datensätzen, das zur Bewertung des endgültigen neuronalen Netzes verwendet wird. Der Fehler für die Holdout-Stichprobe bietet eine "ehrliche" Schätzung der Vorhersagekraft des Modells, da die Holdout-Fälle (die Fälle in der Holdout-Stichprobe) nicht zur Erstellung des Modells verwendet wurden.

- **Fälle auf der Grundlage der relativen Anzahl an Fällen zufällig zuweisen.** Geben Sie die relative Anzahl (Verhältnis) der Fälle an, die den einzelnen Stichproben (Training, Test und Holdout) nach dem Zufallsprinzip zugewiesen werden sollen. Die Spalte **%** gibt auf der Grundlage der von Ihnen angegebenen Werte für die relative Anzahl den Prozentsatz der Fälle an, die den einzelnen Stichproben zugewiesen werden.

Die Angabe von 7, 3, 0 als relative Anzahl für Training-, Test- und Holdout-Stichprobe entspricht 70 %, 30 % und 0 %. Die Angabe von 2, 1, 1 als Werte für die relative Anzahl entspricht 50 %, 25 % und 25 %. 1, 1, 1 entspricht der Aufteilung des Datasets in drei gleich große Teile für Training, Test und Holdout.

- **Partitionsvariable zum Zuweisen von Fällen verwenden.** Geben Sie eine numerische Variable an, die jeden Fall im aktiven Dataset der Trainings-, Test oder Holdout-Stichprobe zuweist. Fälle mit einem positiven Wert für die Variable werden der Trainingsstichprobe zugewiesen, Fälle mit dem Wert 0 der Teststichprobe und Fälle mit einem negativen Wert der Holdout-Stichprobe. Fälle mit einem systemdefiniert fehlenden Wert werden aus der Analyse ausgeschlossen. Alle benutzerdefiniert fehlenden Werte für die Partitionsvariable werden immer als gültig behandelt.

## Architektur

Auf der Registerkarte "Architektur" können Sie die Struktur des Netzes angeben. Diese Prozedur erstellt ein neuronales Netz mit genau einer verborgenen Schicht vom Typ "Radiale Basisfunktion". Normalerweise ist es nicht erforderlich, diese Einstellungen zu ändern.

**Anzahl der Einheiten in der verborgenen Schicht.** Es gibt drei Möglichkeiten zur Auswahl der Anzahl der verborgenen Einheiten.

1. **Beste Anzahl an Einheiten innerhalb eines automatisch berechneten Bereichs finden.** Die Prozedur berechnet automatisch den Mindest- und Maximalwert des Bereichs und ermittelt die beste Anzahl an verborgenen Einheiten innerhalb des Bereichs.

Wenn eine Teststichprobe definiert wurde, verwendet die Prozedur das Testdatenkriterium: Die beste Anzahl an verborgenen Einheiten ist diejenige, die den kleinsten Fehler in den Testdaten ergibt. Wenn keine Teststichprobe definiert wurde, verwendet die Prozedur das Bayes-Informationskriterium (BIC): Die beste Anzahl an verborgenen Einheiten ist diejenige, die auf der Basis der Trainingsdaten den kleinsten BIC ergibt.

2. **Beste Anzahl an Einheiten innerhalb eines angegebenen Bereichs finden.** Sie können selbst einen Bereich angeben und die Prozedur ermittelt die "beste" Anzahl an verborgenen Einheiten innerhalb dieses Bereichs. Wie zuvor wird auch hier die beste Anzahl an verborgenen Einheiten mithilfe des Testdatenkriteriums oder des Bayes-Informationskriteriums (BIC) ermittelt.
3. **Eine vorgegebene Anzahl an Einheiten verwenden.** Sie können die Verwendung eines Bereichs außer Kraft setzen und stattdessen direkt eine bestimmte Anzahl an Einheiten eingeben.

**Aktivierungsfunktion für verborgene Schicht.** Die Aktivierungsfunktion für die verborgene Schicht ist die radiale Basisfunktion, die die Einheiten in einer Schicht mit den Werten der Einheiten in der vorhergehenden Schicht "verknüpft". Bei der Ausgabeschicht dient die Identitätsfunktion als Aktivierungsfunktion. Die Ausgabeinheiten sind also einfach gewichtete Summen der verborgenen Einheiten.

- **Normalisierte radiale Basisfunktion.** Verwendet die Aktivierungsfunktion "Softmax", sodass die Aktivierungen aller verborgenen Einheiten so normalisiert werden, dass ihre Summe 1 ergibt.
- **Gewöhnliche radiale Basisfunktion.** Verwendet die exponentielle Aktivierungsfunktion, sodass die Aktivierung der verborgenen Einheit eine Gaußglocke als Funktion der Eingaben darstellt.

**Überschneidung zwischen versteckten Einheiten.** Der Überschneidungsfaktor ist ein Multiplikator, der auf die Breite der radialen Basisfunktionen angewendet wird. Der automatisch berechnete Wert des Überschneidungsfaktors lautet  $1+0,1d$ , wobei  $d$  die Anzahl der Eingabeeinheiten ist (die Summe aus der Anzahl an Kategorien in allen Faktoren und der Anzahl der Kovariaten).

## Ausgabe

**Netzstruktur.** Zeigt zusammenfassende Informationen über das neuronale Netz an.

- **Beschreibung.** Zeigt Informationen über das neuronale Netz an, einschließlich der abhängigen Variablen, der Anzahl der Ein- und Ausgabeinheiten, der Anzahl der verdeckten Ebenen und Einheiten und der Aktivierungsfunktionen.
- **Diagramm.** Zeigt das Netzdiagramm als nicht bearbeitbares Diagramm an. Beachten Sie: Mit steigender Anzahl an Kovariaten und Faktorstufen wird das Diagramm schwerer zu interpretieren.
- **Synaptische Gewichtungen.** Zeigt die Koeffizientenschätzungen an, die die Beziehung zwischen den Einheiten in einer bestimmten Schicht und den Einheiten in der nächsten Schicht anzeigen. Die synaptischen Gewichtungen beruhen auf der Trainingsstichprobe, selbst wenn das aktive Dataset in Trainings-, Test- und Holdout-Daten partitioniert ist. Beachten Sie, dass die Anzahl der synaptischen Gewichtungen recht groß werden kann und dass diese Gewichtungen im Allgemeinen nicht zur Interpretation der Netzergebnisse verwendet werden.

**Netzleistung.** Zeigt die Ergebnisse an, die verwendet werden, um zu bestimmen, ob das Modell "gut" ist. *Hinweis:* Die Diagramme in dieser Gruppe beruhen auf der Kombination aus Trainings- und Teststichprobe bzw. nur auf der Trainingsstichprobe, wenn keine Teststichprobe vorhanden ist.

- **Modellzusammenfassung.** Zeigt eine Zusammenfassung der Ergebnisse des neuronalen Netzes nach Partition und insgesamt an, einschließlich der folgenden Werte: Fehler, Relativer Fehler oder Prozentsatz der falschen Vorhersagen und Trainingszeit.

Der Fehler ist der Quadratsummenfehler. Außerdem werden die relativen Fehler oder Prozentsätze der falschen Vorhersagen in Abhängigkeit von den Messniveaus der abhängigen Variablen angezeigt. Wenn eine abhängige Variable ein metrisches Messniveau aufweist, wird der durchschnittliche relative Gesamtfehler (relativ zum Mittelwertmodell) angezeigt. Wenn alle abhängigen Variablen kategorial sind, wird der durchschnittliche Prozentsatz der falschen Vorhersagen angezeigt. Die relativen Fehler

oder Prozentsätze der falschen Vorhersagen werden jeweils für die einzelnen abhängigen Variablen angezeigt.

- **Klassifikationsergebnisse.** Zeigt für jede kategoriale abhängige Variable eine Klassifikationstabelle an. Jede Tabelle gibt für jede Kategorie abhängiger Variablen die Anzahl der korrekt und nicht korrekt klassifizierten Fälle an. Der Prozentsatz der Gesamtzahl der Fälle, die korrekt klassifiziert wurden, wird ebenfalls angegeben.
- **ROC-Kurve.** Zeigt eine ROC-Kurve (Receiver Operating Characteristic) für jede kategoriale abhängige Variable an. Außerdem wird eine Tabelle angezeigt, die die Fläche unter den einzelnen Kurven angibt. Bei jeder abhängigen Variablen zeigt das ROC-Diagramm jeweils genau eine Kurve für jede Kategorie an. Wenn die abhängige Variable zwei Kategorien aufweist, behandelt jede Kurve die fragliche Kategorie als positiven Zustand gegenüber der anderen Kategorie. Wenn die abhängige Variable mehr als zwei Kategorien aufweist, behandelt jede Kurve die fragliche Kategorie als positiven Zustand gegenüber allen anderen Kategorien.
- **Kumulative Gewinndiagramme.** Zeigt für jede kategoriale abhängige Variable ein kumulatives Gewinndiagramm an. Die Anzeige einer Kurve für jede Kategorie der abhängigen Variablen verhält sich wie bei ROC-Kurven.
- **Liftdiagramm.** Zeigt für jede kategoriale abhängige Variable ein Liftdiagramm an. Die Anzeige einer Kurve für jede Kategorie der abhängigen Variablen verhält sich wie bei ROC-Kurven.
- **Diagramm "Vorhergesagt/Beobachtet".** Zeigt für jede abhängige Variable ein Diagramm an, das die vorhergesagten Werte in Abhängigkeit von den beobachteten Werten angibt. Bei kategorialen abhängigen Variablen werden für jede Antwortkategorie gruppierte Boxplots der vorhergesagten Pseudowahrscheinlichkeiten angezeigt, wobei die Kategorie der beobachteten Antworten als Clustervariable fungiert. Bei metrischen abhängigen Variablen wird ein Streudiagramm angezeigt.
- **Diagramm "Residuum/Vorhergesagt".** Zeigt für jede metrische abhängige Variable ein Diagramm an, das die Residuen in Abhängigkeit von den vorhergesagten Werten angibt. Es sollte kein Muster zwischen Residuen und vorhergesagten Werten zu beobachten sein. Dieses Diagramm wird nur bei metrischen abhängigen Variablen erstellt.

**Zusammenfassung der Fallverarbeitung.** Zeigt die Tabelle mit der Zusammenfassung der Fallverarbeitung an, die die Anzahl der in der Analyse ein- und ausgeschlossenen Fälle zusammenfasst (insgesamt und nach Trainings-, Test- und Holdout-Stichprobe geordnet).

**Wichtigkeitsanalyse für unabhängige Variablen.** Führt eine Sensitivitätsanalyse durch, mit der die Wichtigkeit der einzelnen Prädiktoren für die Bestimmung des neuronalen Netzes berechnet wird. Die Analyse beruht auf der Kombination aus Trainings- und Teststichprobe bzw. nur auf der Trainingsstichprobe, wenn keine Teststichprobe vorhanden ist. Dadurch werden eine Tabelle und ein Diagramm erstellt, die die Wichtigkeit und die normalisierte Wichtigkeit für die einzelnen Prädiktoren anzeigen. Beachten Sie, dass die Sensitivitätsanalyse rechenintensiv und zeitaufwendig ist, wenn eine große Anzahl an Prädiktoren oder Fällen vorliegt.

## Speichern

Auf der Registerkarte "Speichern" können Vorhersagen im Dataset als Variablen gespeichert werden.

- **Für jede abhängige Variable vorhergesagten Wert oder Kategorie speichern.** Damit wird bei metrischen abhängigen Variablen der vorhergesagte Wert und bei kategorialen abhängigen Variablen die vorhergesagte Kategorie gespeichert.
- **Für jede abhängige Variable vorhergesagte Pseudowahrscheinlichkeit speichern.** Damit werden bei kategorialen abhängigen Variablen die vorhergesagten Pseudowahrscheinlichkeiten gespeichert. Für die ersten  $n$  Kategorien wird eine separate Variable gespeichert. Dabei wird  $n$  in der Spalte *Zu speichernde Kategorien* angegeben.

**Namen der gespeicherten Variablen.** Durch eine automatische Generierung von Namen wird sichergestellt, dass Ihre Arbeit nicht verloren geht. Mit benutzerdefinierten Namen können Sie Ergebnisse aus früheren Durchgängen verwerfen oder ersetzen, ohne zuerst die gespeicherten Variablen im Dateneditor löschen zu müssen.

Wahrscheinlichkeiten und Pseudowahrscheinlichkeiten

Vorhergesagte Pseudowahrscheinlichkeiten können nicht als Wahrscheinlichkeiten interpretiert werden, da die Prozedur "Radiale Basisfunktion" für die Ausgabeschicht den Quadratsummenfehler und die Aktivierungsfunktion "Identität" verwendet. Die Prozedur speichert diese vorhergesagten Pseudowahrscheinlichkeiten, selbst wenn sie kleiner als 0 oder größer als 1 sind oder wenn die Summe für eine abhängige Variable nicht 1 ergibt.

ROC-Diagramme, kumulative Gewinnindigramme und Liftdiagramme (siehe „Ausgabe“ auf Seite 15) werden auf der Basis von Pseudowahrscheinlichkeiten erstellt. Für den Fall, dass eine der Pseudowahrscheinlichkeiten kleiner als 0 oder größer als 1 ist oder die Summe für eine gegebene Variable nicht 1 ist, werden sie zunächst auf einen Wert zwischen 0 und 1 und auf die Summe 1 skaliert. Pseudowahrscheinlichkeiten werden durch Division durch ihre Summe skaliert. Wenn ein Fall beispielsweise für eine abhängige Variable mit drei Kategorien vorhergesagte Pseudowahrscheinlichkeiten von 0,50, 0,60 und 0,40 aufweist, wird jede Pseudowahrscheinlichkeit durch die Summe 1,50 dividiert, woraus sich die Werte 0,33, 0,40 und 0,27 ergeben.

Wenn negative Pseudowahrscheinlichkeiten vorliegen, werden vor der oben beschriebenen Neuskalierung alle Pseudowahrscheinlichkeiten jeweils mit dem Betrag der niedrigsten Wahrscheinlichkeit addiert. Wenn die Pseudowahrscheinlichkeiten beispielsweise  $-0,30$ ,  $0,50$  und  $1,30$  betragen, müssen Sie zunächst  $0,30$  zu jedem Wert addieren und erhalten somit die Werte  $0,00$ ,  $0,80$  und  $1,60$ . Als Nächstes dividieren Sie die einzelnen neuen Werte durch die Summe  $2,40$ , wodurch sich die Werte  $0,00$ ,  $0,33$  und  $0,67$  ergeben.

## Export

Die Registerkarte "Export" dient zum Speichern der Schätzungen der synaptischen Gewichtungen für die einzelnen abhängigen Variablen in einer XML-Datei (PMML-Datei). Anhand dieser Modelldatei können Sie die Modellinformationen zu Scoring-Zwecken auf andere Datendateien anwenden. . Diese Option ist nicht verfügbar, wenn aufgeteilte Dateien definiert wurden.

## Optionen

**Benutzerdefiniert fehlende Werte.** Faktoren müssen gültige Werte für einen Fall aufweisen, um in die Analyse aufgenommen zu werden. Mit diesen Steuerelementen legen Sie fest, ob benutzerdefiniert fehlende Werte bei den Faktoren und kategorialen abhängigen Variablen als gültige Werte behandelt werden sollen.



## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. IBM stellt dieses Material möglicherweise auch in anderen Sprachen zur Verfügung. Für den Zugriff auf das Material in einer anderen Sprache ist eine Kopie des Produkts oder der Produktversion in der jeweiligen Sprache erforderlich.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Director of Licensing*

*IBM Europe, Middle East & Africa*

*North Castle Drive, MD-NC119*

*Armonk, NY 10504-1785*

*France*

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*19-21, Nihonbashi-Hakozakicho, Chuo-ku*

*Tokyo 103-8510, Japan*

Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. -->

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter als IBM werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen

Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

*IBM Director of Licensing*

*IBM Europe, Middle East & Africa*

*North Castle Drive, MD-NC119*

*Armonk, NY 10504-1785*

*France*

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die angeführten Leistungsdaten und Kundenbeispiele dienen nur zur Illustration. Die tatsächlichen Ergebnisse beim Leistungsverhalten sind abhängig von der jeweiligen Konfiguration und den Betriebsbedingungen.

Alle Informationen zu Produkten anderer Anbieter als IBM stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussage zu Leistung, Kompatibilität und anderen Merkmalen von Produkten anderer Anbieter als IBM machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter als IBM sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmieretechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme ohne Zahlung an IBM kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© Copyright IBM Corp. 2021. Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet.

© Copyright IBM Corp. 1989 - 2021. All rights reserved.

## Marken

---

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corp in den USA und/ oder anderen Ländern. Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Her-



stellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.



---

# Index

## A

- Abbruchregeln
  - in "Mehrschichtiges Perzeptron" [11](#)
- Aktivierungsfunktion
  - in "Mehrschichtiges Perzeptron" [6](#)
  - in "Radiale Basisfunktion" [14](#)
- Architektur
  - Neuronale Netze [2](#)
- Ausgabeschicht
  - in "Mehrschichtiges Perzeptron" [6](#)
  - in "Radiale Basisfunktion" [14](#)

## B

- Batch-Training
  - in "Mehrschichtiges Perzeptron" [7](#)

## F

- fehlende Werte
  - in "Mehrschichtiges Perzeptron" [11](#)

## G

- Gewinndiagramm
  - in "Mehrschichtiges Perzeptron" [9](#)
  - in "Radiale Basisfunktion" [15](#)

## H

- Holdout-Stichprobe
  - in "Mehrschichtiges Perzeptron" [5](#)
  - in "Radiale Basisfunktion" [14](#)

## L

- Liftdiagramm
  - in "Mehrschichtiges Perzeptron" [9](#)
  - in "Radiale Basisfunktion" [15](#)

## M

- Mehrschicht-Perzeptron
  - output [9](#)
- Mehrschichtiges Perzeptron
  - Modellexport [11](#)
  - Netzarchitektur [6](#)
  - Optionen [11](#)
  - Partitionen [5](#)
  - Speichern von Variablen im aktiven Dataset [10](#)
  - Training [7](#)
- Mini-Batch-Training
  - in "Mehrschichtiges Perzeptron" [7](#)

## N

- Netzarchitektur
  - in "Mehrschichtiges Perzeptron" [6](#)
  - in "Radiale Basisfunktion" [14](#)
- Netzdiagramm
  - in "Mehrschichtiges Perzeptron" [9](#)
  - in "Radiale Basisfunktion" [15](#)
- Netztraining
  - in "Mehrschichtiges Perzeptron" [7](#)
- Neuronale Netze
  - Architektur [2](#)

## O

- Online-Training
  - in "Mehrschichtiges Perzeptron" [7](#)

## R

- Radiale Basisfunktion
  - Modellexport [17](#)
  - Netzarchitektur [14](#)
  - Optionen [17](#)
  - output [15](#)
  - Partitionen [14](#)
  - Speichern von Variablen im aktiven Dataset [16](#)
- ROC-Kurve
  - in "Mehrschichtiges Perzeptron" [9](#)
  - in "Radiale Basisfunktion" [15](#)

## T

- Teststichprobe
  - in "Mehrschichtiges Perzeptron" [5](#)
  - in "Radiale Basisfunktion" [14](#)
- Trainingsstichprobe
  - in "Mehrschichtiges Perzeptron" [5](#)
  - in "Radiale Basisfunktion" [14](#)

## V

- Verborgene Schicht
  - in "Mehrschichtiges Perzeptron" [6](#)
  - in "Radiale Basisfunktion" [14](#)





