IBM SPSS Neural Networks 30

注

本書および本書で紹介する製品をご使用になる前に、17 ページの『特記事項』に記載されている情報をお読みください。

製品情報

この版は、30 のバージョン IBM® SPSS®Statistics のリリース 0、修正 0、および新しい版で特に指示がない限り、それ以降のすべてのリリースと修正に適用されます。

目次

第 1 章 ニューラル・ネットワーク

以下のニューラル・ネットワーク機能が、SPSSStatistics Premium Edition または Neural Networks オプションに含まれています。

ニューラル・ネットワークの概要

ニューラル・ネットワークは、その機能、柔軟性、使いやすさのため、多くの予測データ・マイニング・アプリケーションに推奨されるツールです。 予測ニューラル・ネットワークは、基礎となるプロセスが複雑なアプリケーションで特に役に立ちます。例を次に示します。

- 生産および配送コストの合理化に対する消費者の需要を予測する。
- ダイレクト・メールによるマーケティングへの応答の確率を予測し、メーリング・リストに掲載されているどの世帯にオファーを送るかを判断する。
- 申請者をスコアリングし、貸付限度拡大のリスクを判断する。
- 保険金請求データベース内の不正なトランザクションを検出する。

多層パーセプトロン (MLP) ネットワークや放射基底関数 (RBF) ネットワークなど、予測アプリケーションで使用されるニューラル・ネットワークは、モデル予測結果を対象変数の既知の値と比較できるという意味で監視されます。 Neural Networks オプションを使用すると、MLP ネットワークと RBF ネットワークを適合させ、結果のモデルをスコアリングのために保存できます。

ニューラル・ネットワークとは

ニューラル・ネットワークという用語は、脳機能の研究に由来し、大きなパラメーター領域と柔軟な構造を特徴とする、大まかな関連性のあるモデル・ファミリーに適用されます。 関連する用語の多くはその起源を反映していますが、ファミリーが大きくなるにつれ、新しいモデルの大半は非生物学的なアプリケーション用に設計されてきました。

ニューラル・ネットワークの具体的な定義は、それらが使用される分野と同様にさまざまです。 モデルのファミリー全体を正しくカバーすることはできませんが、現時点では、以下の説明を検討してください。[1]:

ニューラル・ネットワークは、その性質上、経験に基づいた知識を保存して利用できるようになる傾向を持つ大規模な並列分散プロセッサーです。 ニューラル・ネットワークは、次の 2 点において脳と似ています。

- 知識はネットワークによって、学習プロセスを介して獲得されます。
- シナプスの重みとして知られるニューロン間の結合強度を使用して、知識が保存されます。

なぜこの定義が制限すぎる理由については、以下を参照してください。[2].

この定義によってニューラル・ネットワークと従来の統計的手法とを区別するには、説明されていない 事柄が定義の実際の文章と同じように重要です。 例えば、従来の線型回帰モデルでは、最小二乗法を使用して知識を獲得し、その知識を回帰係数に保存できます。 この意味では、これはニューラル・ネットワークです。 実際、線型回帰は特定のニューラル・ネットワークの特別な事例であるといえます。 ただし、線型回帰では、厳格なモデル構造と、データから学習する前に適用される一連の仮定を使用します。

それに対し、前述の定義では、モデル構造と仮定に関する要求は最小限です。 したがって、従属変数と独立変数の間の特定の関係を事前に仮定することなく、ニューラル・ネットワークを幅広い統計モデルに近似させることができます。 代わりに、関係の形式は学習プロセス時に決定されます。 従属変数および独立変数間の線型関係が適切な場合、ニューラル・ネットワークの結果は線型回帰モデルの結果に近似します。

---

1 Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation*, 2nd ed. New York: Macmillan College Publishing.

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

非線型の関係がより適切である場合、ニューラル・ネットワークは自動的に「適切な」モデル構造を概算します。

この柔軟性に対する代償として、ニューラル・ネットワークのシナプスの重みを容易に解釈することはできません。 つまり、従属変数と独立変数の関係を生成する基礎のプロセスを説明しようとする場合、従来の統計モデルを使用する方が良いことがあります。 ただし、モデルの解釈のしやすさが重要でない場合は、ニューラル・ネットワークを使用する方が、多くの場合、迅速に優れたモデル結果を取得できます。

ニューラル・ネットワークの構造

ニューラル・ネットワークは、モデル構造や仮定に対する要求が最小限ですが、一般的な**ネットワーク・アーキテクチャー**を理解するのに役立ちます。 多層パーセプトロン (MLP) ネットワークや放射基底関数 (RBF) ネットワークはターゲット変数 (出力とも呼ばれる) の予測誤差を最小にする予測 (入力または独立変数とも呼ばれる) の関数です。

製品に付属している *bankloan.sav* データ・セットについて考察してみます。データ・セットから、融資申請者のプールから潜在的な債務不履行者を識別できる必要があります。 この問題に適用される MLP ネットワークまたは RBF ネットワークは、債務不履行を予測する場合の誤差を最小にする測定の関数です。 次の図は、この関数の形式の関連付けに役に立ちます。

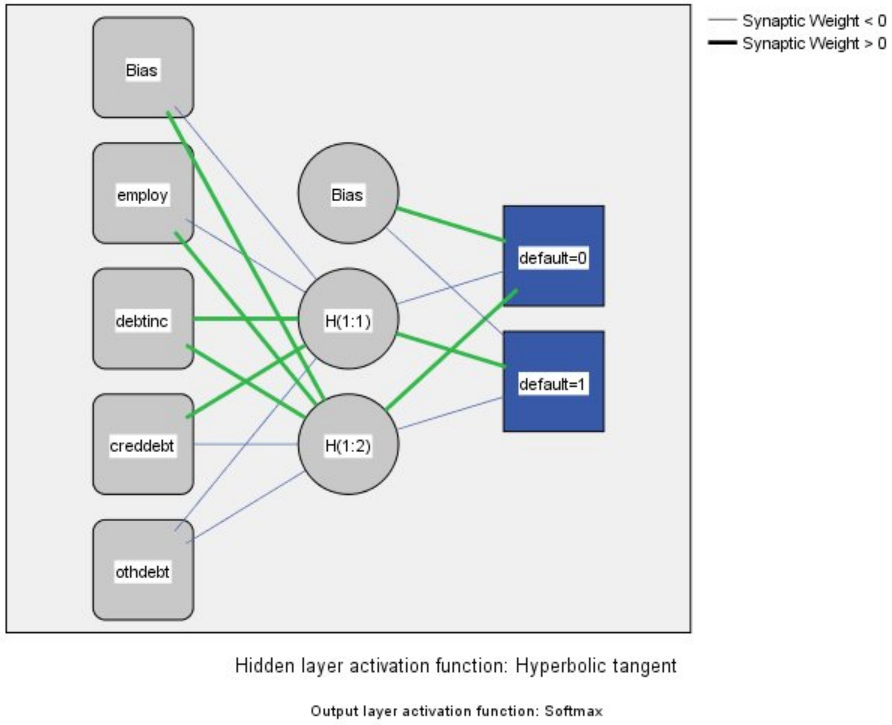

図 *1. 1* つの隠れ層を含むフィードフォワード・アーキテクチャー

ネットワーク内の接続は、フィードバック・ループなしで入力層から出力層へ順方向にフローするため、この構造は**フィードフォワード・アーキテクチャー**として知られています。 この図について、次に説明します。

- **入力層**には予測値が含まれます。
- **隠れ層**には、観測不可能なノードまたはユニットが含まれます。各隠れユニットの値は、何らかの予測値の関数です。正確な関数の形式は、一部はネットワークのタイプに、また一部はユーザーが制御可能な指定によって決まります。
- **出力層**には、応答が含まれます。 債務不履行の履歴は 2 つのカテゴリーを持つカテゴリー変数であるため、2 つの指示変数として記録されます。 各出力ユニットは、隠れユニットの何らかの関数です。 ここでも、正確な関数の形式は、一部はネットワークのタイプに、また一部はユーザーが制御可能な指定によって決まります。

MLP ネットワークでは、2 番目の隠れ層が許可されます。その場合、2 番目の隠れ層の各ユニットは、最初の隠れ層のユニットの関数で、各応答は 2 番目の隠れ層のユニットの関数です。

多層パーセプトロン

多層パーセプトロン (MLP) プロシージャーは、予測変数の値に基づいて、1 つ以上の従属 (ターゲット) 変数の予測モデルを生成します。

例。 MLP プロシージャーを使用した 2 つのシナリオを次に示します。

銀行の融資担当者は、債務不履行になる可能性がある人物を示す特徴を識別し、それらの特徴を使用して、信用リスクの良し悪しを識別できる必要があります。 過去の顧客のサンプルを使用して、マルチレイヤー・パーセプトロンをトレーニングし、過去の顧客のホールドアウト・サンプルを使用して分析を検証してから、ネットワークを使用して、見込み顧客を信用リスクとしてグッドまたはバッドかどうかを分類することができます。

病院のシステムでは、心筋梗塞 (MI または「心臓発作」) の治療で入院している患者の入院費用と期間の追跡に関心があります。 これらの指標の正確な推定値を取得することで、患者が治療を受けるときに、管理者が空きベッド数を適切に管理することができます。MI の治療を受けた患者のサンプルの処理記録を利用して、管理者はネットワークをトレーニングして、コストと滞在期間の両方を予測することができる。

データの考慮事項

従属変数。 従属変数には次のものを使用できます。

- *Nominal* (名義). 変数の値が固有のランキングのないカテゴリー (例えば、従業員が勤務する会社の部門) を表す場合は、その変数を名義型として扱うことができます。 名義変数の例としては、 地域、郵便番号、宗教上の所属などが挙げられます。
- *Ordinal* (順序データ). 変数の値が、いくつかの固有のランキングを持つカテゴリーを表している場合 (例えば、サービス満足度のレベルが「非常に不満」から「非常に満足」まで)、その変数を序数として扱うことができます。 順序変数の例としては、 満足度や信頼度を表す得点や嗜好得点などが挙げられます。
- スケール *(scale)*. 変数の値が有意な測定基準を持つ順序付きカテゴリーを表す場合、その変数をスケール (連続型) として扱うことができます。これにより、値の間の距離の比較が適切になります。 スケール変数の例としては、 年齢や、千ドル単位で表した所得が挙げられます。

  このプロシージャーは適切な測定レベルがすべての従属変数に割り当てられていることを想定しています。ただし、ソース変数リスト内の変数を右クリックし、ポップアップ・メニューから測定レベルを選択して、変数の測定レベルを一時的に変更することができます。

変数リストで各変数の隣にあるアイコンは、以下のような尺度とデータ型を表します。

表 *1.* 測定の尺度のアイコン

	数値	**ストリング**	**日付**	**時間**
スケール (連続)		該当しない		
順序型				
名義				

予測変数: 予測値は、因子 (カテゴリー) または共変量 (スケール) として指定することができます。

カテゴリー変数のコード化: この手続きは、手続きの期間について、one-of-*c* コード化を使用してカテゴリー予測変数と従属変数を一時的に再割り当てします。 変数の *c* カテゴリーが存在する場合、その変数は c ベクトルとして格納され、最初のカテゴリー (1,0,...,0)、次のカテゴリー (0,1,0,...,0) などのように、最後のカテゴリー (0,0,...,0,1) までが示されます。

このコード化方式ではシナプスの重みの数が増加するため、学習が遅くなる可能性があります。ただし、より「コンパクトな」コード化方法では、多くの場合ニューラル・ネットワークがあまり適切でなくなります。ネットワーク・トレーニングが非常にゆっくり進行している場合は、類似のカテゴリーを結合するか、または極端に少ないカテゴリーを持つケースをドロップすることによって、カテゴリー予測変数のカテゴリー数を削減してみてください。

検定サンプルまたはホールドアウト・サンプルが定義されている場合でも、one-of-*c* コード化はすべて学習データに基づいています (5 ページの『データ区分 』を参照)。 そのため、検定サンプルまたはホールドアウト・サンプルに学習データに存在しない予測カテゴリーを持つケースが含まれる場合、それらのケースはプロシージャーでもスコアリングでも使用されません。 検定サンプルまたはホールドアウト・サンプルに学習データに存在しない従属変数カテゴリーを持つケースが含まれる場合、それらのケースはプロシージャーでは使用されませんが、スコアリングされることがあります。

再調整: スケール従属変数および共変量は、ネットワーク学習の向上のため、デフォルトで再スケールされます。 検定サンプルまたはホールドアウト・サンプルが定義されている場合でも、再スケールはすべて学習データに基づいて行われます (5 ページの『データ区分 』を参照)。 つまり再スケールのタイプに応じて、共変量または従属変数の平均値、標準偏差、最小値、または最大値が学習データのみを使用して計算されます。 変数を指定してデータ区分を定義する場合、これらの共変量または従属変数の学習サンプル、検定サンプル、ホールドアウト・サンプル全体への分布が類似していることが重要です。

度数による重み付け: 度数による重み付けは、この手続きでは無視されます。

結果の再現: 結果を正確に再現する場合、同じプロシージャーの設定を使用するだけでなく、乱数ジェネレーターに同じ初期化値、同じデータの順序、同じ変数の順序を使用します。 詳細について、以下に示します。

- **乱数の生成 (Random number generation)**: このプロシージャーでは、データ区分のランダムな割り当て時に乱数生成、シナプスの重みの初期化にランダムなサブサンプリング、自動アーキテクチャー選択にランダムなサブサンプリング、および重みの初期化と自動アーキテクチャー選択で使用されるシミュレーテッド・アニーリング・アルゴリズムを使用します。 今後同じランダム化された結果が再現されるようにするためには、多層パーセプトロン・プロシージャーの実行前に、毎回乱数ジェネレーターに同じ初期化値を使用してください。 のトピックを参照してください。
- **ケースの並び順:** オンライン学習方法およびミニバッチ学習方法 (7 ページの『学習 』を参照) は、ケースの順序に明示的に依存しますが、シナプスの重みの初期化ではデータ・セットからのサブサンプリングが含まれるため、バッチ学習方法もケースの順序に依存します。

  並び順の影響を最小限に抑えるには、ケースを無作為に並べます。 特定の解の安定性を確認するには、異なる無作為な順序でソートされたケースを使用していくつかの異なる解を取得します。 ファイル・サイズが非常に大きい場合は、異なる無作為な順序でソートされたケースのサンプルを使用し、複数回に分けて実行することができます。
- **変数の表示順**: 変数の順序が変更されると、さまざまなパターンの初期値が割り当てられるため、因子および共変量リストの変数の順序に結果が影響を受けることがあります。 ケースの順序の影響と同じように、さまざまな変数の順序を試して (因子および共変量リスト内でドラッグ・アンド・ドロップするだけ)、特定の解の安定性を評価できます。

多層パーセプトロン・ネットワークの作成

メニューから次の項目を選択します。

「**分析**」 > 「**ニューラル ネットワーク**」 > 「**多層パーセプトロン...**」

1. 1 つ以上の従属変数を選択します。
2. 少なくとも 1 つの因子または共変量を選択します。

オプションで、「変数」タブで共変量を再スケールする方法を変更できます。 次の項目から選択します。

- **標準化**: 平均値を減算し、標準偏差で除算します。つまり、(*x*−mean)/*s* です。
- **正規化**: 最小値を減算し、範囲で除算します。つまり、(*x*−min)/(max−min) です。 正規化された値は 0 から 1 の間に収まります。
- **調整済み正規化**: 最小値を減算し、範囲で除算した値を調整したものです。つまり、[2*(*x*−min)/(max−min)]−1 です。 調整済みの正規化された値は −1 から 1 の間に収まります。
- **なし**: 共変量を再スケールしません。

不明な尺度のフィールド

データ・セット内の 1 つ以上の変数 (フィールド) の測定レベルが不明な場合、測定レベルの警告が表示されます。 測定レベルはこの手続きの結果の計算に影響を与えるため、すべての変数について測定レベルを定義する必要があります。

データをスキャン: アクティブ・データ・セットのデータを読み込み、デフォルトの測定レベルを、測定レベルが現在不明なすべてのフィールドに割り当てます。 データ・セットのサイズが大きい場合、この処理には時間がかかります。

手動で割り当てる: 不明な測定レベルを持つフィールドをすべて表示するダイアログが開きます。 このダイアログを使用して、測定レベルをこれらのフィールドに割り当てることができます。 データ・エディターの「変数ビュー」でも、測定レベルを割り当てることができます。

この手続きでは測定レベルが重要であるため、すべてのフィールドに対して測定レベルが定義されるまで、ダイアログにアクセスしてこの手続きを実行することはできません。

データ区分

「**データ区分データ・セット**」。 このグループは、アクティブ・データ・セットを学習サンプル、検定サンプル、およびホールドアウト・サンプルに区分する方法を指定します。 **学習サンプル**はニューラル・ネットワークによる学習に使用されるデータ・レコードから構成されます。モデルを取得するために、データ・セット内の一定の割合のケースを学習サンプルに割り当てる必要があります。 **検定サンプル**は、過学習を防ぐために、学習中の誤差の追跡に使用される、独立したデータ・レコードのセットです。 検定サンプルを作成することを強く推奨します。検定サンプルが学習サンプルより小さい場合、一般的にネットワーク学習が最も効率的になります。 **ホールドアウト・サンプル**は、最終のニューラル・ネットワークを評価するために使用される、もう 1 つの独立したデータ・レコードのセットです。モデルの構築にホールドアウト・ケースが使用されていないため、ホールドアウト・サンプルの誤差によって、モデルの予測能力を「公正に」推定できます。

- 「**ケースの相対値に基づいてランダムにケースを割り当てる**」。 各サンプル (学習、検定、ホールドアウト) にランダムに割り当てられるケースの相対数 (比率) を指定します。 **%** 列には、指定した相対数に基づいて各サンプルに割り当てられるケースのパーセントが表示されます。

  例えば、学習サンプル、検定サンプル、ホールドアウト・サンプルの相対数として 7、3、0 を指定すると、それぞれ 70%、30%、0% になります。 相対数として 2、1、1 を指定すると、それぞれ 50%、25%、25% になります。1、1、1 と指定すると、データ・セットは学習サンプル、検定サンプル、ホールドアウト・サンプルにそれぞれ 3 分の 1 ずつ分けられます。
- 「**データ区分変数を使用してケースを割り当てる**」。 アクティブ・データ・セットの各ケースを学習サンプル、検定サンプル、またはホールドアウト・サンプルに割り当てる数値変数を指定します。 変数に正の値を持つケースは学習サンプルに、0 の値を持つケースは検定サンプルに、負の値を持つケースはホールドアウト・サンプルに割り当てられます。 システム欠損値を持つケースは、分析から除外されます。 分割変数のユーザー欠損値は、常に有効な値として扱われます。

注: データ区分変数を使用すると、プロシージャーを連続して実行しても同一の結果は保証されません。『多層パーセプトロン』の『結果の再現』を参照してください。

アーキテクチャー

「アーキテクチャー」タブを使用して、ネットワークの構造を指定します。 このプロシージャーでは、自動的に「最適な」アーキテクチャーを選択するか、カスタム・アーキテクチャーを指定できます。

自動アーキテクチャー選択では、1 つの隠れ層を持つネットワークが構築されます。 隠れ層に許可されるユニットの最小数と最大数を指定すると、自動アーキテクチャー選択によって、隠れ層に「最適な」ユニットの数が計算されます。 自動アーキテクチャー選択では、隠れ層および出力層に対してデフォルトのアクティブ化関数が使われます。

カスタム・アーキテクチャー選択では、隠れ層および出力層の専門的な制御が可能で、必要なアーキテクチャーが事前にわかっている場合や自動アーキテクチャー選択の結果を調整する必要がある場合に最も役立つことがあります。

「隠れ層」

隠れ層には、観測不可能なネットワーク・ノード (ユニット) が含まれています。 それぞれの隠れ単位は、入力の加重合計の関数です。 この関数は活性化関数であり、 重みの値は推定アルゴリズムによって決定されます。 ネットワークに 2 番目の隠れ層が含まれている場合、その 2 番目の層にあるそれぞれの隠れ層は、最初の隠れ層の単位の加重合計の関数です。 同じアクティブ化関数が、両方の層で使用されます。

「隠れ層の数」。多層パーセプトロンは 1 つまたは 2 つの隠れ層を持つことができます。

*Activation Function (*活性化関数*).* 活性化関数は、層の単位の加重合計を後続の層の単位の値に「リンク」します。

- 「**双曲線正接**」。 この関数の形式は、$\gamma(c) = \tanh(c) = (e^{c}-e^{-c})/(e^{c}+e^{-c})$ です。 これは実数の引数を取り、それらを範囲 (–1, 1) に変換します。 自動アーキテクチャー選択を使用した場合、これは隠れ層のすべてのユニットに対するアクティブ化関数となります。
- 「**S 字曲線**」。 この関数の形式は、$\gamma(c) = 1/(1+e^{-c})$ です。 これは実数の引数を取り、それらを範囲 (0, 1) に変換します。

「単位数」。各隠れ層の単位数は、 明示的に指定するか、推定アルゴリズムによって自動的に決定することができます。

「出力層」

出力層には、ターゲット (従属) 変数が含まれます。

*Activation Function (*活性化関数*).* 活性化関数は、層の単位の加重合計を後続の層の単位の値に「リンク」します。

- 「**同一**」。 この関数の形式は $\gamma(c) = c$ です。 これは実数の引数を取り、それらを変換せずに返します。 自動アーキテクチャー選択を使用した場合、これは、スケール従属変数のある出力層のユニットに対するアクティブ化関数となります。
- 「**ソフトマックス**」。 この関数の形式は、$\gamma(c_k) = \exp(c_k)/\Sigma_j\exp(c_j)$ です。 これは実数の引数のベクトルを取り、それを、要素が範囲 (0, 1) に収まり、合計が 1 となるベクトルに変換します。 ソフトマックスは、すべての従属変数がカテゴリー型である場合にのみ使用できます。 自動アーキテクチャー選択を使用した場合、これは、すべての従属変数がカテゴリー型である出力層のユニットに対するアクティブ化関数となります。
- 「**双曲線正接**」。 この関数の形式は、$\gamma(c) = \tanh(c) = (e^{c}-e^{-c})/(e^{c}+e^{-c})$ です。 これは実数の引数を取り、それらを範囲 (–1, 1) に変換します。
- 「**S 字曲線**」。 この関数の形式は、$\gamma(c) = 1/(1+e^{-c})$ です。 これは実数の引数を取り、それらを範囲 (0, 1) に変換します。

「**スケール従属変数の再スケール**」。 これらのコントロールは、少なくとも 1 つのスケール従属変数が選択されている場合にのみ使用できます。

- 「**標準化**」。 平均値を減算し、標準偏差で除算します。つまり、$(x-\text{mean})/s$ です。
- **正規化**: 最小値を減算し、範囲で除算します。つまり、$(x-\text{min})/(\text{max}-\text{min})$ です。 正規化された値は 0 から 1 の間に収まります。 出力層が S 字曲線アクティブ化関数を使用している場合、スケール従属変数に対して必要な再スケール方法です。 訂正オプションでは、再スケール式への訂正として適用される小さな数 ε を指定します。この訂正により、再スケールされたすべての従属変数の値は、アクティブ化関数の範囲内となります。 特に、x が最小値および最大値を取る場合に未訂正の式で発生する 0 と 1 の値は、S 字曲線関数の範囲の限界を定義しますが、その範囲内ではありません。 訂正された式は $[x-(\text{min}-\varepsilon)]/[(\text{max}+\varepsilon)-(\text{min}-\varepsilon)]$ となります。 0 以上の数値を指定します。
- 「**調整済み正規化**」。 最小値を減算し、範囲で除算した値を調整したものです。つまり、$[2*(x-\text{min})/(\text{max}-\text{min})]-1$ です。 調整済みの正規化された値は –1 から 1 の間に収まります。 出力層が双曲線正接アクティブ化関数を使用している場合、スケール従属変数に対して必要な再スケール方法です。 訂正オプションでは、再スケール式への訂正として適用される小さな数 ε を指定します。この訂正により、再スケールされたすべての従属変数の値は、アクティブ化関数の範囲内となります。 特に、x が最小値および最大値を取る場合に未訂正の式で発生する -1 と 1 の値は、双曲線正接関数の範囲の限界を定義しますが、その範囲内ではありません。 訂正された式は $\{2*[(x-(\text{min}-\varepsilon))/((\text{max}+\varepsilon)-(\text{min}-\varepsilon))]\}-1$ となります。 0 以上の数値を指定します。
- **なし :** スケール従属変数の再スケールはありません。

学習

「学習」タブを使用して、ネットワークの学習方法を指定します。 学習のタイプと最適化アルゴリズムによって、使用できる学習オプションが決定します。

「**学習のタイプ**」。 学習のタイプによって、ネットワークがレコードをどのように処理するかが決定します。次の学習のタイプからいずれかを選択します。

- 「**バッチ**」。 すべての学習データ・レコードをパスした後にのみ、シナプスの重みを更新します。つまり、バッチ学習では学習データ・セットのすべてのレコードの情報を使用します。 バッチ学習は、全体の誤差を直接最小化するため、多くの場合に推奨されます。ただし、バッチ学習は、いずれかの停止規則に一致するまで、何回も重みの更新が必要になる場合があるため、多くのデータ・パスが必要になる場合があります。 バッチ学習は「小さな」データ・セットで最も役立ちます。
- 「**オンライン**」。 1 つ 1 つの学習データ・レコードの後で、シナプスの重みを更新します。つまり、オンライン学習では一度に 1 つのレコードの情報を使用します。 オンライン学習では継続的にレコードを取得し、いずれかの停止規則に一致するまで重みを更新します。 すべてのレコードが 1 回使用され、一致する停止規則がない場合、データ・レコードを再利用して処理が継続されます。オンライン・トレーニングは、関連付けられた予測値を持つ「より大きい」データ・セットのバッチよりも優れています。つまり、多くのレコードと多くの入力が存在し、それらの値が互いに独立していない場合、オンライン・トレーニングではバッチ学習よりも合理的な応答をより迅速に得ることができます。
- 「**ミニバッチ**」。 学習データ・レコードをほぼ等しいサイズのグループに分割し、1 つのグループをパスした後でシナプスの重みを更新します。つまり、ミニバッチ学習ではレコードのグループの情報を使用します。 この処理では、必要に応じてデータ・グループを再利用します。 ミニバッチ学習はバッチ学習とオンライン学習の中間に位置し、「中規模サイズの」データ・セットの場合に最適です。 このプロシージャーでは、ミニバッチあたりの学習レコード数を自動的に判断するか、または 1 より大きく、メモリーに格納するケースの最大数以下の整数を指定することができます。 メモリーに格納するケースの最大数は「オプション」タブで指定できます。

「**最適化アルゴリズム**」。 この方法は、シナプスの重みを推定するために使用します。

- 「**スケーリングされた共役勾配**」。共役勾配法の使用を正当化する前提は、バッチ・トレーニング・タイプにのみ適用されるため、この方法は、オンライン・トレーニングまたはミニバッチ学習には使用できません。
- 「**勾配降下**」。この方法は、オンライン・トレーニングまたはミニバッチ学習で使用する必要があります。バッチ学習で使用することもできます。

「**学習オプション**」。 学習オプションを使用すると、最適化アルゴリズムを微調整できます。 通常、ネットワークで推定の問題が発生しない限り、これらの設定を変更する必要はありません。

スケーリングされた共役勾配アルゴリズムの学習オプションには次のものがあります。

- 「**初期ラムダ**」。 スケーリングされた共役勾配アルゴリズムのラムダ・パラメーターの初期値です。 0 より大きく 0.000001 より小さい数を指定します。
- 「**初期シグマ**」。 スケーリングされた共役勾配アルゴリズムのシグマ・パラメーターの初期値です。 0 より大きく 0.0001 より小さい数を指定します。
- 「**区間の中心**」および「**区間のオフセット**」。 区間の中心 (α_0) と区間のオフセット (α) は、シミュレーテッド・アニーリングが使用された場合、重みのベクトルがランダムに生成される区間 $[\alpha_0-\alpha, \alpha_0+\alpha]$ を定義します。 シミュレーテッド・アニーリングは、最適化アルゴリズムの適用時にグローバルな最小値を検出する目的で、ローカルの最小値から抜け出すために使用します。 このアプローチは、重みの初期化と自動アーキテクチャー選択に使用します。 区間の中心には数値を指定し、区間のオフセットには 0 より大きい数値を指定してください。

勾配降下アルゴリズムの学習オプションには次のものがあります。

- 「**初期の学習率**」。 勾配降下アルゴリズムの学習率の初期値です。 学習率が高いと、ネットワークの学習は短時間になりますが、不安定になるという犠牲を伴う可能性があることを意味します。 0 より大きい数値を指定します。
- 「**学習率の下限**」。 勾配降下アルゴリズムの学習率の下限です。 この設定は、オンライン学習およびミニバッチ学習にのみ適用されます。 0 より大きく、初期の学習率より小さい数を指定します。

- 「**運動量**」。 勾配降下アルゴリズムの初期運動量パラメーターです。 運動量の項によって、高すぎる学習率によって発生する不安定性を回避するのに役立ちます。 0 より大きい数値を指定します。
- 「**学習率の減衰 (エポック)**」。 オンライン学習またはミニバッチ学習で勾配降下が使用される場合、初期の学習率を学習率の下限まで減少させるために必要な学習サンプルのエポック数 (*p*) またはデータ・パス数です。 これにより学習率の減衰因子 $\beta = (1/p\ K)*\ln(\eta_0/\eta_{low})$ を制御できます。ここで、η_0 は初期の学習率を表し、η_{low} は学習率の下限、*K* は学習データ・セット内のミニバッチの総数 (またはオンライン学習の場合、学習レコード数) です。 0 より大きい整数を指定します。

出力

「**ネットワーク構造**」。 ニューラル・ネットワークの要約情報を表示します。

- 「**説明**」。ニューラル・ネットワークに関する情報を表示します。この情報には、従属変数、入出力装置の数、非表示の層と装置の数、および活動化機能が含まれます。
- 「**図**」。 ネットワーク図を編集不可能なグラフとして表示します。 共変量の数と因子レベルが増加すると、図の解釈が難しくなることに注意してください。
- 「**シナプスの重み**」。 特定の層のユニットと次の層のユニットとの関係を示す係数の推定値を表示します。 アクティブなデータ・セットが学習データ、検定データ、ホールドアウト・データに区分されている場合でも、シナプスの重みは学習サンプルに基づいています。 シナプスの重みの数はかなり多くなる可能性があり、通常これらの重みはネットワーク結果の解釈には使用されないことに注意してください。

「**ネットワーク・パフォーマンス**」。 モデルが「適している」かどうかを判断するために使用する結果を表示します。 注: このグループのグラフは、学習サンプルと検定サンプルの組み合わせか、または検定サンプルがない場合は学習サンプルのみに基づいています。

- **モデルの要約:** 誤差、相対誤差または誤った予測のパーセント、学習の停止に使用される停止規則、および学習時間など、データ区分別および全体のニューラル・ネットワークの結果の要約を表示します。

  同一、S 字曲線、双曲線正接のアクティブ化関数が出力層に適用される場合、この誤差は平方和の誤差です。 ソフトマックス・アクティブ化関数が出力層に適用される場合、クロスエントロピー誤差となります。

  相対誤差または誤った予測のパーセントは、従属変数の測定レベルに応じて表示されます。 従属変数にスケール測定レベルがある場合、全体の相対誤差の平均 (平均モデルに相対的) が表示されます。 すべての従属変数がカテゴリー型の場合、誤った予測の平均のパーセントが表示されます。 相対誤差または誤った予測のパーセントは、個々の従属変数に対しても表示されます。
- 「**分類結果**」。 各カテゴリー従属変数の分類テーブルをデータ区分別と全体で表示します。 各テーブルは、各従属変数カテゴリーに正しくまたは誤って分類されたケースの数を示します。 正しく分類されたすべてのケースのパーセントも報告されます。
- 「**ROC 曲線**」。 各カテゴリー従属変数の ROC (受信者動作特性) 曲線を表示します。 また、各曲線の下の領域を示すテーブルも表示します。 特定の従属変数に対し、ROC グラフはカテゴリーごとに 1 つの曲線を表示します。 従属変数に 2 つのカテゴリーがある場合、各曲線は問題のカテゴリーを他方のカテゴリーに対する正の状態として扱います。 従属変数に 2 つを超えるカテゴリーがある場合、各曲線は問題のカテゴリーをその他すべてのカテゴリーの集計に対する正の状態として扱います。
- **累積ゲイン グラフ:** 各カテゴリー従属変数の累積ゲイン・グラフを表示します。 ROC 曲線と同様、従属変数カテゴリーごとに 1 つの曲線を表示します。
- **リフト (インデックス) グラフ:** 各カテゴリー従属変数のリフト・グラフを表示します。 ROC 曲線と同様、従属変数カテゴリーごとに 1 つの曲線を表示します。
- 「**予測対観測のグラフ**」。 各従属変数の予測値と観測値のグラフを表示します。 カテゴリー従属変数の場合、予測された疑似確率のクラスター箱ひげ図が、観測された各応答カテゴリーをクラスター変数として、応答カテゴリーごとに表示されます。 スケール従属変数の場合、散布図が表示されます。
- **残差と予測グラフ:** 各スケール従属変数の残差と予測値のグラフを表示します。 残差と予測値の間に明らかなパターンはないはずです。 このグラフはスケール従属変数に対してのみ生成されます。

「**処理したケースの要約**」。 分析に含まれたケースおよび分析から除外されたケースの数を、全体と、学習サンプル、検定サンプル、およびホールドアウト・サンプルごとに要約するケース処理要約テーブルを表示します。

「**独立変数の重要度分析**」。 感度分析を実行し、ニューラル・ネットワークの決定における各予測値の重要度を計算します。 分析は、学習サンプルと検定サンプルの組み合わせか、または検定サンプルがない場合は学習サンプルのみに基づきます。 この分析によって、各予測値の重要度および正規化された重要度を表すテーブルおよびグラフを作成します。 大量の予測値またはケースがある場合、感度分析の計算は効率的でなく時間もかかります。

保存

「保存」タブを使用して、データ・セットに変数として予測を保存します。

- 「**従属変数ごとの予測値または予測カテゴリーを保存**」。 これにより、スケール従属変数の予測値を保存し、カテゴリー従属変数の予測カテゴリーを保存します。
- 「**従属変数ごとの予測擬似確率を保存、またはカテゴリ従属変数に予測擬似確率を保存**」。 カテゴリー従属変数の予測疑似確率を保存します。 最初の *n* 個のカテゴリーのそれぞれに、個別のカテゴリー変数が保存されます。*n* は「**保存するカテゴリー**」列で指定されます。

「**保存された変数の名前**」。 名前の自動生成を使用すると、すべての作業が保存されます。 カスタム名を使用すると、データ・エディターで保存された変数を最初に削除することなく、前回実行された結果を破棄または置き換えることができます。

確率および疑似確率

ソフトマックス・アクティブ化およびクロスエントロピー誤差を含むカテゴリー従属変数には、各カテゴリーの予測値が含まれます。この場合、各予測値はケースがカテゴリーに属する確率です。

平方和の誤差を含むカテゴリー従属変数には各カテゴリーの予測値が含まれますが、予測値は確率として解釈できません。 いずれかの予測疑似確率が 0 未満または 1 より大きい場合であっても、または特定の従属変数の合計が 1 でない場合でも、このプロシージャーではこれらの予測疑似確率を保存します。

ROC、累積ゲイン・グラフ、およびリフト・グラフ (8 ページの『出力 』を参照) は、疑似確率に基づいて作成されます。疑似確率のいずれかが 0 より小さいか、 1 より大きいか、与えられた変数の合計が 1 でない場合、最初の再呼び出しは 0 から 1 の間であり、 1 に合計されます。 疑似確率はその合計で除算することによって再スケールされます。 例えば、ケースで 3 つのカテゴリー従属変数に対し 0.50、0.60、および 0.40 の予測疑似確率が存在する場合、各疑似確率は合計の 1.50 で除算され、0.33、0.40、0.27 になります。

疑似確率のいずれかが負の場合、前述の再スケールを行う前に最も低い確率の絶対値がすべての疑似確率に加算されます。 例えば、疑似確率が -0.30、0.50、および 1.30 である場合、まず各値に 0.30 を加算して 0.00、0.80、および 1.60 を算出します。 次に、それぞれの新しい値を、合計の 2.40 で除算すると、0.00、0.33、および 0.67 となります。

エクスポート

「エクスポート」タブを使用して、各従属変数のシナプスの重みの推定値を XML (PMML) ファイルに保存します。 このモデル・ファイルを使用して、スコアリングのために他のデータ・ファイルにモデル情報を適用できます。 のトピックを参照してください。 分割ファイルが定義されている場合、このオプションは使用できません。

オプション

「**ユーザー欠損値**」。 因子は、分析に含めるケースで有効な値を取る必要があります。 これらのコントロールによって、ユーザー欠損値を因子およびカテゴリー従属変数で有効な値として扱うかどうかを決定できます。

「**停止規則**」。 これらは、ニューラル・ネットワークの学習を停止するタイミングを決定する規則です。 学習は、少なくとも 1 つのデータ・パスで行われます。 学習は、以下の基準に従って停止でき、ここに示された順序で確認されます。 下記の停止規則の定義において、ステップはオンラインおよびミニバッチ方式ではデータ・パスに、バッチ方式では反復に対応します。

- 「**エラーの減少なしの最大ステップ数**」。 誤差の減少を確認する前に許可されるステップ数です。 指定されたステップ数を経て誤差の減少がない場合、学習が停止します。 0 より大きい整数を指定します。 また、エラーの計算に使用されたデータ・サンプルを指定することもできます。 「**自動選択**」では、検定

サンプルが存在する場合はそれを使用し、存在しない場合は学習サンプルを使用します。 バッチ学習は各データ・パスの後の学習サンプルの誤差の減少を保証するため、検定サンプルが存在する場合、このオプションはバッチ学習に対してのみ適用されます。 「**学習とテスト データ**」では、これらの各サンプルの誤差を確認します。このオプションは検定サンプルが存在する場合にのみ適用されます。

注: それぞれの完全なデータ・パスの後、オンライン学習とミニバッチ学習には、学習誤差を計算するために追加のデータ・パスが必要です。 この追加のデータ・パスでは、学習の速度がかなり遅くなる場合があります。そのため、一般にどのケースでも検定サンプルを提供して、「**自動選択**」を選択することをお勧めします。

- 「**最大学習時間**」。 アルゴリズムを実行する最大時間 (分) を指定するかどうかを選択します。 0 より大きい数値を指定します。
- 「**最大学習エポック数**」。 許可される最大エポック (データ・パス) 数です。 エポックの最大数を超えた場合、学習が停止します。 0 より大きい整数を指定します。
- 「**学習エラーの最小相対変化**」。 以前のステップと比較した学習誤差の相対変化が基準の値を下回った場合、学習は停止します。 0 より大きい数値を指定します。オンライン・トレーニングおよびミニバッチ・トレーニングの場合、エラーの計算にテスト・データのみが使用される場合、この基準は無視されます。
- 「**学習エラー率の最小相対変化**」。 帰無仮説モデルの誤差に対する学習誤差の割合が基準値を下回った場合、学習は停止します。 帰無仮説モデルはすべての従属変数に対する平均値を予測します。 0 より大きい数値を指定します。オンライン・トレーニングおよびミニバッチ・トレーニングの場合、エラーの計算にテスト・データのみが使用される場合、この基準は無視されます。

「**メモリーに格納する最大ケース数**」。 これにより、多層パーセプトロン・アルゴリズム内の次の設定を制御します。 1 より大きい整数を指定します。

- 自動アーキテクチャー選択では、ネットワーク・アーキテクチャーの決定に使用されるサンプルのサイズは min(1000,*memsize*) です。ここで、*memsize* はメモリーに格納するケースの最大数です。
- ミニバッチの数を自動的に計算するミニバッチ学習では、ミニバッチの数は min(max(*M*/10,2),*memsize*) です。ここで、*M* は学習サンプルのケース数です。

放射基底関数

放射基底関数 (RBF) プロシージャーは、予測変数の値に基づいて、1 つ以上の従属 (ターゲット) 変数の予測モデルを生成します。

例。 ある通信プロバイダーは、サービス利用パターンによって顧客ベースを区分し、顧客を 4 つのグループに分類しました。 人口統計学データを使用してグループ・メンバーシップを予測する RBF ネットワークは、企業が個々の見込み顧客に対するオファーをカスタマイズできるようにします。

データの考慮事項

従属変数。 従属変数には次のものを使用できます。

- *Nominal (名義)*. 変数の値が固有のランキングのないカテゴリー (例えば、従業員が勤務する会社の部門) を表す場合は、その変数を名義型として扱うことができます。 名義変数の例としては、 地域、郵便番号、宗教上の所属などが挙げられます。
- *Ordinal (順序データ)*. 変数の値が、いくつかの固有のランキングを持つカテゴリーを表している場合 (例えば、サービス満足度のレベルが「非常に不満」から「非常に満足」まで)、その変数を序数として扱うことができます。 順序変数の例としては、 満足度や信頼度を表す得点や嗜好得点などが挙げられます。
- スケール *(scale)*. 変数の値が有意な測定基準を持つ順序付きカテゴリーを表す場合、その変数をスケール (連続型) として扱うことができます。これにより、値の間の距離の比較が適切になります。 スケール変数の例としては、 年齢や、千ドル単位で表した所得が挙げられます。

  このプロシージャーは適切な測定レベルがすべての従属変数に割り当てられることを想定しています。ただし、ソース変数リスト内の変数を右クリックし、ポップアップ・メニューから測定レベルを選択して、変数の測定レベルを一時的に変更することができます。

変数リストで各変数の隣にあるアイコンは、以下のような尺度とデータ型を表します。

表 2. 測定の尺度のアイコン				
	数値	**ストリング**	**日付**	**時間**
スケール (連続)		該当しない		
順序型				
名義				

予測変数。 予測値は、因子 (カテゴリー) または共変量 (スケール) として指定することができます。

カテゴリー変数のコード化。 この手続きは、手続きの期間について、one-of-*c* コード化を使用してカテゴリー予測変数と従属変数を一時的に再割り当てします。 変数の *c* カテゴリーがある場合、その変数は c ベクトルとして保管されます。最初のカテゴリーは (1,0, ..., 0)、次のカテゴリーは (0,1,0, ..., 0)、..., 最終カテゴリー (0,0, ..., 0, 1)。

このコード化方式ではシナプスの重みの数が増加するため、学習が遅くなる可能性があります。しかし、より「コンパクトな」コード化方法では、多くの場合ニューラル・ネットワークがあまり適切でなくなります。 ネットワーク・トレーニングが非常にゆっくり進行している場合は、類似のカテゴリーを結合するか、または極端に少ないカテゴリーを持つケースをドロップすることによって、カテゴリー予測変数のカテゴリー数を削減してみてください。

検証サンプルまたはホールドアウト・サンプルが定義されている場合でも、one-of-*c* コード化はすべて学習データに基づいています (12 ページの『データ区分』を参照)。 そのため、検証サンプルまたはホールドアウト・サンプルに学習データに存在しない予測カテゴリーを持つケースが含まれる場合、それらのケースはプロシージャーでもスコアリングでも使用されません。 検証サンプルまたはホールドアウト・サンプルに学習データに存在しない従属変数カテゴリーを持つケースが含まれる場合、それらのケースはプロシージャーでは使用されませんが、スコアリングされることがあります。

再調整: スケール従属変数および共変量は、ネットワーク学習の向上のため、デフォルトで再スケールされます。 検証サンプルまたはホールドアウト・サンプルが定義されている場合でも、再スケールはすべて学習データに基づいて行われます (12 ページの『データ区分』を参照)。 つまり再スケールのタイプに応じて、共変量または従属変数の平均値、標準偏差、最小値、または最大値が学習データのみを使用して計算されます。 変数を指定してデータ区分を定義する場合、これらの共変量または従属変数の学習サンプル、検証サンプル、ホールドアウト・サンプル全体への分布が類似していることが重要です。

度数による重み付け: 度数による重み付けは、この手続きでは無視されます。

結果の再現。 結果を正確に再現する場合、同じプロシージャーの設定を使用するだけでなく、乱数ジェネレーターに同じ初期化値と、同じデータの順序を使用します。 詳細について、以下に示します。

- **乱数の生成**。 このプロシージャーでは、データ区分のランダムな割り当て時に乱数生成を使用します。今後同じランダム化された結果が再現されるようにするためには、放射基底関数プロシージャーの実行前に、毎回乱数ジェネレーターに同じ初期化値を使用してください。
- **ケースの並び順:** 放射基底関数を決定するために、2 段階のクラスター・アルゴリズムが使われるため、結果はデータの順序にも依存します。

  並び順の影響を最小限に抑えるには、ケースを無作為に並べます。 特定の解の安定性を確認するには、異なる無作為な順序でソートされたケースを使用していくつかの異なる解を取得します。 ファイル・サイズが非常に大きい場合は、異なる無作為な順序でソートされたケースのサンプルを使用し、複数回に分けて実行することができます。

放射基底関数ネットワークの作成

メニューから次の項目を選択します。

「**分析**」 > 「**ニューラル ネットワーク**」 > 「**放射基底関数...**」

1. 1 つ以上の従属変数を選択します。

2. 少なくとも 1 つの因子または共変量を選択します。

オプションで、「変数」タブで共変量を再スケールする方法を変更できます。 次の項目から選択します。

- **標準化:** 平均値を減算し、標準偏差で除算します。つまり、(*x*–mean)/*s* です。
- **正規化済み**。 最小値を減算し、範囲で除算します。つまり、(*x*–min)/(max–min) です。 正規化された値は 0 から 1 の間に収まります。
- **調整済み正規化**。 最小値を減算し、範囲で除算した値を調整したものです。つまり、[2*(*x*–min)/(max–min)]–1 です。 調整済みの正規化された値は –1 から 1 の間に収まります。
- **なし**。 共変量を再スケールしません。

不明な尺度のフィールド

データ・セット内の 1 つ以上の変数 (フィールド) の測定レベルが不明な場合、測定レベルの警告が表示されます。 測定レベルはこの手続きの結果の計算に影響を与えるため、すべての変数について測定レベルを定義する必要があります。

データをスキャン: アクティブ・データ・セットのデータを読み込み、デフォルトの測定レベルを、測定レベルが現在不明なすべてのフィールドに割り当てます。 データ・セットのサイズが大きい場合、この処理には時間がかかります。

手動で割り当てる: 不明な測定レベルを持つフィールドをすべて表示するダイアログが開きます。 このダイアログを使用して、測定レベルをこれらのフィールドに割り当てることができます。 データ・エディターの「変数ビュー」でも、測定レベルを割り当てることができます。

この手続きでは測定レベルが重要であるため、すべてのフィールドに対して測定レベルが定義されるまで、ダイアログにアクセスしてこの手続きを実行することはできません。

データ区分

「**データ区分データ・セット**」。 このグループは、アクティブ・データ・セットを学習サンプル、検定サンプル、およびホールドアウト・サンプルに区分する方法を指定します。 **学習サンプル**はニューラル・ネットワークによる学習に使用されるデータ・レコードから構成されます。モデルを取得するために、データ・セット内の一定の割合のケースを学習サンプルに割り当てる必要があります。 **検定サンプル**は、過学習を防ぐために、学習中の誤差の追跡に使用される、独立したデータ・レコードのセットです。 検定サンプルを作成することを強く推奨します。検定サンプルが学習サンプルより小さい場合、一般的にネットワーク学習が最も効率的になります。 **ホールドアウト・サンプル**は、最終のニューラル・ネットワークを評価するために使用される、もう 1 つの独立したデータ・レコードのセットです。モデルの構築にホールドアウト・ケースが使用されていないため、ホールドアウト・サンプルの誤差によって、モデルの予測能力を「公正に」推定できます。

- 「**ケースの相対値に基づいてランダムにケースを割り当てる**」。 各サンプル (学習、検定、ホールドアウト) にランダムに割り当てられるケースの相対数 (比率) を指定します。 **%** 列には、指定した相対数に基づいて各サンプルに割り当てられるケースのパーセントが表示されます。

  例えば、学習サンプル、検定サンプル、ホールドアウト・サンプルの相対数として 7、3、0 を指定すると、それぞれ 70%、30%、0% になります。 相対数として 2、1、1 を指定すると、それぞれ 50%、25%、25% になります。1、1、1 と指定すると、データ・セットは学習サンプル、検定サンプル、ホールドアウト・サンプルにそれぞれ 3 分の 1 ずつ分けられます。
- 「**データ区分変数を使用してケースを割り当てる**」。 アクティブ・データ・セットの各ケースを学習サンプル、検定サンプル、またはホールドアウト・サンプルに割り当てる数値変数を指定します。 変数に正の値を持つケースは学習サンプルに、0 の値を持つケースは検定サンプルに、負の値を持つケースはホールドアウト・サンプルに割り当てられます。 システム欠損値を持つケースは、分析から除外されます。 分割変数のユーザー欠損値は、常に有効な値として扱われます。

アーキテクチャー

「アーキテクチャー」タブを使用して、ネットワークの構造を指定します。 このプロシージャーでは、1 つの隠れた「放射基底関数」層を持つニューラル・ネットワークを作成します。通常、これらの設定を変更する必要はありません。

「**隠れ層の単位数**」。 隠れユニットの数を選択するには、3 つの方法があります。

1. **自動的に計算された範囲内の最適なユニット数を見つける**。 このプロシージャーでは、範囲の最小値および最大値を自動的に計算し、範囲内で最適な隠れユニットの数を見つけます。

   検定サンプルが定義されている場合、このプロシージャーでは、検定データ基準 (隠れユニットの最適な数は、検定データでの誤差が最も小さい数) を使用します。 検定サンプルが定義されていない場合、このプロシージャーでは、ベイズ情報量基準 (BIC) (隠れユニットの最適な数は、学習データに基づく BIC が最も小さい数) を使用します。
2. **指定された範囲内の最適なユニット数を見つける**。 独自の範囲を指定し、プロシージャーでその範囲内の「最適な」隠れユニットの数を見つけることができます。 前述と同様、その範囲の隠れユニットの最適な数は、検定データ基準または BIC を使用して決定されます。
3. **指定したユニット数を使用**。 範囲の使用を無効にし、特定のユニットの数を直接指定することができます。

「**隠れ層のアクティブ化関数**」。 隠れ層のアクティブ化関数は放射基底関数で、ある層のユニットを後続の層のユニットの値に「リンク」させます。 出力層の場合、アクティブ化関数は同一関数です。そのため、出力ユニットは隠れユニットの単純に重み付けされた合計です。

- 「**正規化された放射基底関数**」。 ソフトマックス・アクティブ化関数を使用するため、すべての隠れユニットのアクティブ化の合計が 1 に正規化されます。
- 「**通常の放射基底関数**」。 指数アクティブ化関数を使用するため、隠れユニットのアクティブ化は入力の関数としてガウスの「バンプ」となります。

「**隠れ単位の重なり**」。 重なり因子は、放射基底関数の幅に適用される乗数です。 重なり因子の自動的に計算された値は 1+0.1*d* です。*d* は入力ユニットの数を表します (すべての因子のカテゴリー数と共変量の数の合計)。

出力

ネットワーク構造: ニューラル・ネットワークの要約情報を表示します。

- 「**説明**」。ニューラル・ネットワークに関する情報を表示します。この情報には、従属変数、入出力装置の数、非表示の層と装置の数、および活動化機能が含まれます。
- **ダイアグラム**: ネットワーク図を編集不可能なグラフとして表示します。 共変量の数と因子レベルが増加すると、図の解釈が難しくなることに注意してください。
- **シナプスの重み**: 特定の層のユニットと次の層のユニットとの関係を示す係数の推定値を表示します。 アクティブなデータ・セットが学習データ、検定データ、ホールドアウト・データに区分されている場合でも、シナプスの重みは学習サンプルに基づいています。 シナプスの重みの数はかなり多くなる可能性があり、通常これらの重みはネットワーク結果の解釈には使用されないことに注意してください。

ネットワーク パフォーマンス: モデルが「適している」かどうかを判断するために使用する結果を表示します。 注: このグループのグラフは、学習サンプルと検定サンプルの組み合わせか、または検定サンプルがない場合は学習サンプルのみに基づいています。

- **モデルの要約**: 誤差、相対誤差または誤った予測のパーセント、および学習時間など、データ区分別および全体のニューラル・ネットワークの結果の要約を表示します。

  誤差は平方和の誤差です。 さらに、相対誤差または誤った予測のパーセントは、従属変数の測定レベルに応じて表示されます。 従属変数にスケール測定レベルがある場合、全体の相対誤差の平均 (平均モデルに相対的) が表示されます。 すべての従属変数がカテゴリー型の場合、誤った予測の平均のパーセントが表示されます。 相対誤差または誤った予測のパーセントは、個々の従属変数に対しても表示されます。
- **分類結果**: 各カテゴリー従属変数の分類テーブルを表示します。 各テーブルは、各従属変数カテゴリーに正しくまたは誤って分類されたケースの数を示します。 正しく分類されたすべてのケースのパーセントも報告されます。
- **ROC 曲線**: 各カテゴリー従属変数の ROC (受信者動作特性) 曲線を表示します。 また、各曲線の下の領域を示すテーブルも表示します。 特定の従属変数に対し、ROC グラフはカテゴリーごとに 1 つの曲線を表示します。 従属変数に 2 つのカテゴリーがある場合、各曲線は問題のカテゴリーを他方のカテゴリーに対する正の状態として扱います。 従属変数に 2 つを超えるカテゴリーがある場合、各曲線は問題のカテゴリーをその他すべてのカテゴリーの集計に対する正の状態として扱います。

- **累積ゲイン グラフ**: 各カテゴリー従属変数の累積ゲイン・グラフを表示します。 ROC 曲線と同様、従属変数カテゴリーごとに 1 つの曲線を表示します。
- **リフト (インデックス) グラフ**: 各カテゴリー従属変数のリフト・グラフを表示します。 ROC 曲線と同様、従属変数カテゴリーごとに 1 つの曲線を表示します。
- **予測と観測グラフ**: 各従属変数の予測値と観測値のグラフを表示します。 カテゴリー従属変数の場合、予測された疑似確率のクラスター箱ひげ図が、観測された各応答カテゴリーをクラスター変数として、応答カテゴリーごとに表示されます。 スケール従属変数の場合、散布図が表示されます。
- **残差と予測グラフ**: 各スケール従属変数の残差と予測値のグラフを表示します。 残差と予測値の間に明らかなパターンはないはずです。 このグラフはスケール従属変数に対してのみ生成されます。

ケース処理の要約: 分析に含まれたケースおよび分析から除外されたケースの数を、全体と、学習サンプル、検定サンプル、およびホールドアウト・サンプルごとに要約するケース処理要約テーブルを表示します。

独立変数の重要度分析: 感度分析を実行し、ニューラル・ネットワークの決定における各予測値の重要度を計算します。 分析は、学習サンプルと検定サンプルの組み合わせか、または検定サンプルがない場合は学習サンプルのみに基づきます。 この分析によって、各予測値の重要度および正規化された重要度を表すテーブルおよびグラフを作成します。 大量の予測値またはケースがある場合、感度分析の計算は効率的でなく時間もかかります。

保存

「保存」タブを使用して、データ・セットに変数として予測を保存します。

- 「**従属変数ごとの予測値または予測カテゴリーを保存**」。 これにより、スケール従属変数に予測値を保存し、カテゴリー従属変数に予測カテゴリーを保存します。
- 「**従属変数ごとの予測疑似確率を保存**」。 カテゴリー従属変数の予測疑似確率を保存します。 最初の *n* 個のカテゴリーのそれぞれに、個別のカテゴリー変数が保存されます。*n* は「保存するカテゴリー」列で指定されます。

保存する変数の名前: 名前の自動生成を使用すると、すべての作業が保存されます。 カスタム名を使用すると、データ・エディターで保存された変数を最初に削除することなく、前回実行された結果を破棄または置き換えることができます。

確率および疑似確率

放射基底関数のプロシージャーでは出力層に平方和の誤差および同一アクティブ化関数を使用するため、予測疑似確率を確率として解釈することはできません。 いずれかの予測疑似確率が 0 未満または 1 より大きい場合であっても、または特定の従属変数の合計が 1 でない場合でも、このプロシージャーではこれらの予測疑似確率を保存します。

ROC、累積ゲイン・グラフ、およびリフト・グラフ (13 ページの『出力 』を参照) は、疑似確率に基づいて作成されます。いずれかの疑似確率が 0 未満か 1 より大きい、または特定の変数の合計が 1 でない場合、最初に疑似確率は 0 から 1 の間で、合計が 1 になるよう再スケールされます。 疑似確率はその合計で除算することによって再スケールされます。 例えば、ケースで 3 つのカテゴリー従属変数に対し 0.50、0.60、および 0.40 の予測疑似確率が存在する場合、各疑似確率は合計の 1.50 で除算され、0.33、0.40、0.27 になります。

疑似確率のいずれかが負の場合、前述の再スケールを行う前に最も低い確率の絶対値がすべての疑似確率に加算されます。 例えば、疑似確率が -0.30、0.50、および 1.30 である場合、まず各値に 0.30 を加算して 0.00、0.80、および 1.60 を算出します。 次に、それぞれの新しい値を、合計の 2.40 で除算すると、0.00、0.33、および 0.67 となります。

エクスポート

「エクスポート」タブを使用して、各従属変数のシナプスの重みの推定値を XML (PMML) ファイルに保存します。 このモデル・ファイルを使用して、スコアリングのために他のデータ・ファイルにモデル情報を適用できます。 のトピックを参照してください。 分割ファイルが定義されている場合、このオプションは使用できません。

オプション

「**ユーザー欠損値**」。 因子は、分析に含めるケースで有効な値を取る必要があります。 これらのコントロールによって、ユーザー欠損値を因子およびカテゴリー従属変数で有効な値として扱うかどうかを決定できます。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。 この資料は、IBM から他の言語でも提供されている可能性があります。 ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能を IBM は他の国で提供していない場合があります。 日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。 本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。 これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。 ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。 本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。 実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

IBM Director of Licensing

IBM Corporation

日本アイ・ビー・エム株式会社法務・知的財産知的財産権ライセンス渉外

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Legal and Intellectual Property Law
:NONE.
19-21, Nihonbashi-Hakozakicho, Chuo-ku

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、 商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。 国または地域によっては、法律の強行規定により、保証責任の制限が 禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。 本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。 それらの Web サイトにある資料は、この IBM 製品の資料の一部では ありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、 自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing

IBM Corporation

日本アイ・ビー・エム株式会社法務・知的財産知的財産権ライセンス渉外

本プログラムに関する上記の情報は、適切な使用条件の下で使用すること ができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。 実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM はこれらの製品をテストしていないため、IBM 以外の製品に関連するパフォーマンス、互換性、またはその他のクレームの正確性を確認できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。 より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。 お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。 このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。 従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。 これらのサンプル・プログラムは特定物として現存するままの状態で提供されるものであり、いかなる保証も提供されません。 IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次の ように、著作権表示を入れていただく必要があります。

商標

IBM、IBM ロゴ、および ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。 他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。 現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ナ行]

[ハ行]

[マ行]

[ラ行]

R