Problem Statement

We consider the problem of predicting an target variable y given an n-dimensional object $\mathbf{x}$. The target variable y is assumed to be continuous for regression task and binary for classification task. The goal is to build a model f which outcomes a prediction for each input object. We assume that the function $f(\mathbf{x}|\mathbf{w})$ is a feed-forward dense neural network with 1 hidden, and 1 output layer. The hidden layer dimension is denoted by h. The model output is

$$f(\mathbf{x}|\mathbf{w}) = \sigma_2\left(\mathbf{W}_2\sigma_1(\mathbf{W}_1\mathbf{x})\right),$$

where $\mathbf{W}_1 \in \mathbb{R}^{h\times n}$, $\mathbf{W}_2 \in \mathbb{R}^{1\times h}$ are weight matrices, σ_1, σ_2 are activation functions applied to each input component. We omitted the bias terms for simplicity. Let denote by $\mathbf{w} = (\mathrm{vec}(\mathbf{W}_1), \mathrm{vec}(\mathbf{W}_2)) \in \mathbb{R}^p$ a joint weight vector, where $p = h(n+1)$.

Let assume that we are given the design matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\intercal \in \mathbb{R}^{m\times n}$ and the target vector $\mathbf{y} = [y_1, \ldots, y_m]^\intercal \in \mathbb{R}^m$. Each i-th matrix $\mathbf{X}$ row represents an object which is associated with the i-th vector $\mathbf{y}$ element. The goal is to find the optimal weight vector $\mathbf{w}^*$.

The weights $\mathbf{w}$ are fitted by the minimization of an error function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f). \tag{1}$$

The most common choices for the error function $S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f)$ are

- squared error for regression task:

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \sum_{i=1}^{m} \|y_i - f(\mathbf{x}_i|\mathbf{w})\|^2;$$

- cross-entropy for classification task:

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \sum_{i=1}^{m} \left[y_i \log f(\mathbf{x}_i|\mathbf{w}) + (1-y_i)\log(1 - f(\mathbf{x}_i|\mathbf{w}))\right].$$

The problem 1 could be solved by one of the neural network optimization methods [].

The number of model weights p could be extremely huge. In this case the solution of the problem 1 leads to overfitting. To eliminate this problem we propose to select the subset $\mathcal{A} \subseteq \{1, \ldots, p\}$ of the active weights. The weights which are not active are supposed to be zero. To choose the subset $\mathcal{A}$ from all possible 2^p combinations let introduce a quality criteria $Q(\mathcal{A}|\mathbf{X}, \mathbf{y}, f)$. This function evaluates the quality of a particular active set $\mathcal{A}$. We desire to find the optimal subset $\mathcal{A}^*$ which minimize the function

$$\mathcal{A}^* = \arg\min_{\mathcal{A}\subseteq\{1,\ldots,p\}} Q(\mathcal{A}|\mathbf{X}, \mathbf{y}, f). \tag{2}$$

If the solution $\mathcal{A}^*$ of the 2 is given the next step is to determine the optimal model weights $\mathbf{w}^*$ by solving a problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f), \quad \text{subject to } w_j = 0 \text{ for } j \notin \mathcal{A}^*. \tag{3}$$

Feature selection

To find the optimal subset $\mathcal{A}^*$ we suggest to use the approach similar to the mRMR method for feature selection problem. This method tries to select features which have the minimal redundancy and maximum relevance. To formalise this approach let introduce two functions: Sim and Rel. The former measures the redundancy between features, the latter contains relevances between each feature and target vector. We want to minimize the Sim function and maximize the Rel simultaneously. The mRMR method uses the difference of these functions Sim $-$ Rel to choose the optimal feature subset.

The QPFS method offers the explicit way to construct the functions Sim and Rel. The method minimize the following functional

$$(1-\alpha)\mathbf{z}^\intercal\mathbf{Q}\mathbf{z} - \alpha\mathbf{b}^\intercal\mathbf{z} \to \min_{\substack{\mathbf{z}\in\mathbb{R}^n_+ \\ \|\mathbf{z}\|_1=1}}. \tag{4}$$

This functional is an analogue of the described quality criteria 2. The first term is associated with the Sim function and the second with the Rel. The matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ entries measures the pairwise similarities between features. The vector $\mathbf{b} \in \mathbb{R}^n$ express the similarities between each feature and the target vector. The normalized vector $\mathbf{z}$ shows the importance of each feature. This approach penalizes the dependent features and encourages features relevant to the target. The parameter α allows to control the impact of the Sim and the Rel terms. To find the optimal feature subset the thresholding for $\mathbf{z}$ could be applied.

To measure similarity it was proposed to use the absolute value of sample Pearson correlation coefficient or sample mutual information. The problem 4 is convex if the matrix $\mathbf{Q}$ is positive semidefinite. In general it is not always true. To satisfy this condition one could replace this matrix by $\mathbf{Q}-\lambda_{\min}\mathbf{I}$, where $\lambda_{\min}$ is a minimal eigenvalue of $\mathbf{Q}$.

Model weights selection

We propose the algorithm that extends the quadratic programming methodology to the case of model weights selection. Let suppose that the weights from different layers do not interact. This assumption allows to solve the model weights selection problem separately for each layer. We introduce two vectors $\mathbf{z}_1 \in \mathbb{R}^{hn}_+, \|\mathbf{z}_1\|_1 = 1$ and $\mathbf{z}_2 \in \mathbb{R}^{rh}_+, \|\mathbf{z}_2\|_1 = 1$ for matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ respectively. The components of these vectors express the importance of each individual weight. Let introduce the soft version of the problem 2

$$\mathbf{z}_1^* = \arg\min_{\mathbf{z}_1\in\mathbb{R}^{hn}_+\|\mathbf{z}_1\|_1=1} Q(\mathbf{z}_1|\mathbf{X},\mathbf{y},f)$$
$$\mathbf{z}_2^* = \arg\min_{\mathbf{z}_2\in\mathbb{R}^{h}_+\|\mathbf{z}_2\|_1=1} Q(\mathbf{z}_2|\mathbf{X},\mathbf{y},f).$$

Since the vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ are obtained, we have to recover the optimal active subset $\mathcal{A}^*$. Thresholds d_1 and d_2 are tuned for both layers. If the weight importance larger than corresponding threshold the weight is assumed to be active.

Similarly to the authors of QPFS we use the quadratic quality criteria for each layer

$$Q(\mathbf{z}|\mathbf{X},\mathbf{y},f) = (1-\alpha)\mathbf{z}^\intercal\mathbf{Q}\mathbf{z} - \alpha\mathbf{b}^\intercal\mathbf{z}. \tag{5}$$

The matrix $\mathbf{Q}$ entries estimate the pairwise interactions between weights. The vector $\mathbf{b}$ estimates the influence of each weight to the target variable.

The interactions between weights measured by calculating the similarity function (correlation or mutual information) ...

The vector $\mathbf{b}$ entries estimate the influence of the weights to the target variable. Let approximate the function $f(\mathbf{x}|\mathbf{w})$ at the point $\mathbf{w}$ by its linearization

$$\mathbf{f}(\mathbf{X}|\mathbf{w}+\Delta\mathbf{w}) \approx \mathbf{f}(\mathbf{X}|\mathbf{w}) + \mathbf{J}\cdot\Delta\mathbf{w}, \tag{6}$$

where $\mathbf{J} \in \mathbb{R}^{m\times r}$ is a Jacobian matrix

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_r} \\ \cdots & \cdots & \cdots \\ \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_r} \end{pmatrix}. \tag{7}$$

The j-th element of the vector $\mathbf{b}$ equals the similarity function between the j-th column of the matrix $\mathbf{J}$ and the target vector $\mathbf{y}$.

Algorithm 1

Require: $\mathbf{X}, \mathbf{y}$;
Ensure: $\mathbf{a}, \mathbf{w}$;
1: $\mathbf{a}^0 = (1,\ldots,1)^\intercal$
2: $\mathbf{w}^0 = \arg\min_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X},\mathbf{y},f)$
3: **for** $k = 0,\ldots,K$ **do**
4: $\quad \mathbf{a}^{k+1} = \arg\min_{\mathbf{a}\in\{0,1\}^r} \mathbf{a}^\intercal\mathbf{Q}(\mathbf{w}^k)\mathbf{a} - \mathbf{b}^\intercal(\mathbf{w}^k)\mathbf{a}, \quad$ subject to $\mathbf{a}_{k+1}\odot(1-\mathbf{a}_k) = 0$
5: $\quad \mathbf{w}^{k+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X},\mathbf{y},f), \quad$ subject to $\mathbf{w}\odot(1-\mathbf{a}_{k+1}) = 0$

---

1 Thoughts

- Постановка -> квадратичный критерий -> в случае линейной регрессии ... -> пусть Q, b зависят от параметров -> предлагаем итеративный алгоритм
- Итеративный алгоритм: либо начиная с первого слоя и дальше, а какой первый шаг?
- может столбец матрицы Якоби должен коррелировать с ближайшим нейроном а не выходом