Problem Statement

Let assume that we are given the design matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^{\mathsf{T}} \in \mathbb{R}^{m \times n}$ and the target vector $\mathbf{y} = [y_1, \dots, y_m]^{\mathsf{T}} \in \mathbb{R}^m$. 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 build a model f which predicts the target value y given the object $\mathbf{x}$. We assume that the function f is the feed-forward neural network with H hidden layers. Each layer has the weight matrix $\mathbf{W}_h$. We omit the bias term for simplicity. The input of the h layer is $\mathbf{g}_{h-1}$ and the output is $\mathbf{g}_h$. The output $\mathbf{g}_h$ of the h layer is the result of an activation function applied to the linear combination of weight matrix $\mathbf{W}_h$ and the layer input $\mathbf{g}_{h-1}$

$$\begin{aligned}
&\mathbf{g}_h = \text{activation}_h(\mathbf{W}_h \mathbf{g}_{h-1}), \quad h = 1, \dots, H;\\
&\mathbf{g}_h \in \mathbb{R}^{N_h}, \quad h = 0, \dots, H;\\
&\mathbf{g}_0 = \mathbf{x}; \quad N_0 = n, N_H = 1.
\end{aligned}$$

We denote the vectorized union of the layers weights $\mathbf{W}_h$ by $\mathbf{w}$

$$\mathbf{w} = (\text{vec}(\mathbf{W}_1), \dots, \text{vec}(\mathbf{W}_H))^{\mathsf{T}}. \tag{1}$$

In this notations the model outcome is $f(\mathbf{x}|\mathbf{w})$. The result of applying the function f to the matrix $\mathbf{X}$ is $\mathbf{f}(\mathbf{X}|\mathbf{w}) = (f(\mathbf{x}_1|\mathbf{w}), \dots, f(\mathbf{x}_m|\mathbf{w}))^{\mathsf{T}}$. The weights $\mathbf{w}$ are fitted by the minimization of an error function

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) \to \min_{\mathbf{w} \in \mathbb{R}^r}. \tag{2}$$

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

- squared error for regression task:

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \|\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})\|^2;$$

- cross-entropy for classification task:

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

The problem 2 could be solved by one of the neural network optimization methods.

The number of model weights r could be huge. In this case the solution of the problem 2 leads to overfitting. To eliminate this problem we propose to select the subset $\mathcal{A} \in \{1, \dots, r\}$ of the active weights. The weights which are not active are supposed to be zero. To choose the subset $\mathcal{A}$ from the all possible 2^r combinations let introduce a quality criteria $Q(\mathcal{A}|\mathbf{X}, \mathbf{y})$. This function evaluate the quality of a particular active set $\mathcal{A}$

$$\mathcal{A}^* = \underset{\mathcal{A} \subseteq \{1, \dots, r\}}{\arg\min}\, Q(\mathcal{A}|\mathbf{X}, \mathbf{y}). \tag{3}$$

If the solution of the 5 is given the next step is to determine the optimal model weights for the set $\mathcal{A}^*$

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathbb{R}^r}{\arg\min}\, S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f), \quad \text{subject to } w_j = 0 \text{ for } j \notin \mathcal{A}. \tag{4}$$

Quadratic Quality Criteria

In this paper we propose to use the quadratic quality criteria

$$Q(\mathbf{a}) = \mathbf{a}^{\mathsf{T}}\mathbf{Q}\mathbf{a} - \mathbf{b}^{\mathsf{T}}\mathbf{a} \to \min_{\mathbf{a}\in\{0,1\}^n}, \tag{5}$$

where the binary vector $\mathbf{a} \in \{0,1\}^r$ is an indicator of active weights

$$\mathbf{a} = \{a_j\}_{j=1}^r, \quad \text{where } a_j = \begin{cases} 1, \text{ if } j \in \mathcal{A}; \\ 0, \text{ otherwise }. \end{cases} \tag{6}$$

The function $Q(\mathbf{a})$ is an equivalent form of the quality criteria 3.

In the paper [Katrutsa] the quadratic programming approach 3 was implemented to linear regression problem, where the model $f(\mathbf{x}|\mathbf{w})$ outcome is a linear combination of features

$$f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}. \tag{7}$$

In this case the number of parameters equals to the number of features $r = n$ and the problem of selecting active parameters is equivalent to the feature selection problem.

The authors of [Katrutsa] proposed to feature similarities and feature relevances as the parameters $\mathbf{Q}$ and $\mathbf{b}$. The matrix $\mathbf{Q}$ entries measure the pairwise similarities between features. The vector $\mathbf{b}$ entries measure the relevance of the features to the target variable. The absolute value of correlation could be used to measure these interactions. The main drawback of this approach that the stage of active set selection and the model fitting are distinguish procedures.

Algorithm

In this paper we propose to use the extension of described approach to the non-linear case of the neural networks. We introduce the iterative algorithm to the active set selection. Let assume that we already have a solution $\mathbf{w}$ of the problem 2. Now we determine the $\mathbf{Q}$ matrix and the $\mathbf{b}$ vector in the following way.

The matrix $\mathbf{Q}$ entries estimate the pairwise interactions between weights. We assume that the weights from different layers do not interact. To measure the interaction of two weights $[\mathbf{W}_h]_{i_1 j_1}$ and $[\mathbf{W}_h]_{i_2 j_2}$ we calculate the similarity function between neurons $[\mathbf{g}_{h-1}]_{j_1}$ and $[\mathbf{g}_{h-1}]_{j_2}$ which the weights are connected with.

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

$$\mathbf{f}(\mathbf{X}|\mathbf{w} + \Delta\mathbf{w}) \approx \mathbf{f}(\mathbf{X}|\mathbf{w}) + \mathbf{J} \cdot \Delta\mathbf{w}, \tag{8}$$

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

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_1} & \dots & \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_r} \\ \dots & \dots & \dots \\ \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_1} & \dots & \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_r} \end{pmatrix}. \tag{9}$$

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: $\mathbf{a}, \mathbf{w}$;
$\mathbf{a}^0 = (1, \dots, 1)^\mathsf{T}$
$\mathbf{w}^0 = \arg\min\limits_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f)$
for $k = 0, \dots, K$ do
    $\mathbf{a}^{k+1} = \arg\min\limits_{\mathbf{a}\in\{0,1\}^r} \mathbf{a}^\mathsf{T}\mathbf{Q}(\mathbf{w}^k)\mathbf{a} - \mathbf{b}^\mathsf{T}(\mathbf{w}^k)\mathbf{a}, \quad \text{subject to } \mathbf{a}_{k+1} \odot (1 - \mathbf{a}_k) = 0$
    $\mathbf{w}^{k+1} = \arg\min\limits_{\mathbf{w}\in\mathbb{R}^r} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f), \quad \text{subject to } \mathbf{w} \odot (1 - \mathbf{a}_{k+1}) = 0$
```

Two layers

0.1 Levenberg-Marquardt algorithm

To solve the problem 4 we will use the Levenberg-Marquardt iterative procedure. Let assume that we have some initial vector $\mathbf{w}$. We would like to change the value of $\mathbf{w}$ to $\mathbf{w} + \Delta\mathbf{w}$. To determine the $\Delta\mathbf{w}$ let linearize the function $\mathbf{f}$

$$\mathbf{f}(\mathbf{X}|\mathbf{w} + \Delta\mathbf{w}) \approx \mathbf{f}(\mathbf{X}|\mathbf{w}) + J \cdot \Delta\mathbf{w}, \tag{10}$$

where $J \in \mathbb{R}^{n\times r}$

The value of $\Delta\mathbf{w}$ could be found by solving the linear regression problem.

$$\|\mathbf{y} - \mathbf{f}(\mathbf{x}|\mathbf{w} + \Delta\mathbf{w})\|^2 \to \min_{\Delta\mathbf{w}} \tag{11}$$

$$\|J \cdot \Delta\mathbf{w} - (\mathbf{y} - \mathbf{f}(\mathbf{x}|\mathbf{w}))\|^2 \to \min_{\Delta\mathbf{w}} \tag{12}$$

$$\Delta\mathbf{w} = (J^\mathsf{T} J)^{-1} \cdot J^\mathsf{T} \cdot (\mathbf{y} - \mathbf{f}(\mathbf{x}|\mathbf{w})) \tag{13}$$

1 Thoughts

- Квадратичный критерий вынести из постановки
- 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
- $\mathbf{g}(\mathbf{x}|\mathbf{W})$