Аддитивная регуляризация и ее метапараметры при выборе структуры сетей глубокого обучения*

М. С. Потанин[1], К. О. Вайсер[2], В. В. Стрижов[3]

Аннотация

Ключевые слова: выбор моделей; линейные модели; автокодировщик; нейронные сети; структура; генетический алгоритм

Введение

Предлагается эффективный алгоритм вычисления ошибки при использовании контроля по отдельным объектам.
Задан набор данных :

$$\{(\mathbf{x}_j, y_j), j = 1, \ldots, N\}, \mathbf{x}_j = (x_{1j} \cdots x_{nj})^T \in \mathbb{R}^n, y_j \in \mathbb{R}$$

$$\sum_{j=1}^{N} \mathbf{x}_j = \mathbf{0}, \sum_{j=1}^{N} y_j = 0, \frac{1}{N} \sum_{j=1}^{N} x_{ij}^2 = 1, i \in I = \{1, \ldots, n\},$$

*Работа выполнена при поддержке РФФИ (проекты 19-07-1155, 19-07-0885) и правительства РФ (соглашение 05.Y09.21.0018). Настоящая статья содержит результаты проекта «Статистические методы машинного обучения», выполняемого В рамках реализации Программы Центра компетенций Национальной технологической инициативы «Центр хранения и анализа больших данных», поддерживаемого Министерством науки и высшего образования Российской Федерации по Договору МГУ им. М. В. Ломоносова с Фондом поддержки проектов Национальной технологической инициативы от 11.12.2018 № 13/1251/2018.

[1]Московский физико-технический институт, mark.potanin@phystech.edu

[2]Московский физико-технический институт, vajser.ko@phystech.edu

[3]Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук; Московский физико-технический институт, strijov@phystech.edu

который инициализирует "наивную"эластичную сеть. Задача – минимизировать вектор $\mathbf{a}$, параметры регрессии $\hat{y}(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$:

$$\begin{aligned} J_{\text{NEN}}(\mathbf{a}|\lambda_1,\lambda_2) &= \lambda_2 \sum_{i=1}^{n} a_i^2 + \lambda_1 \sum_{i=1}^{n} |a_i| + \sum_{j=1}^{N} \left(y_j - \sum_{i=1}^{n} a_i x_{ij} \right)^2 \\ &= \lambda_2 \mathbf{a}^T\mathbf{a} + \lambda_1 \|\mathbf{a}\|_1 + (\mathbf{y} - \mathbf{X}\mathbf{a})^T(\mathbf{y} - \mathbf{X}\mathbf{a}) \to \min(\mathbf{a}) \\ \mathbf{y} = (y_1 \cdots y_N) \in \mathbb{R}^N, \mathbf{X} &= (\mathbf{x}_1 \cdots \mathbf{x}_N)^T \ (N \times n) \\ \hat{\mathbf{a}}_{\lambda_1,\lambda_2} &= (\hat{a}_{i,\lambda_1,\lambda_2}, i \in I) = \arg\min J_{\text{NEN}}(\mathbf{a}|\lambda_1,\lambda_2) \in \mathbb{R}^n \end{aligned}$$

Предлагается следующий подход:

$$\begin{aligned} J_{\text{EN}}(\mathbf{a}|\lambda_1,\lambda_2) &= \lambda_2 \textstyle\sum_{i=1}^{n} (a_i - a_i^*)^2 + \lambda_1 \sum_{i=1}^{n} |a_i| + \sum_{j=1}^{N} (y_j - \sum_{i=1}^{n} a_i x_{ij})^2 \\ &= \lambda_2 \left(\mathbf{a} - \tfrac{1}{N}\mathbf{X}^T\mathbf{y}\right)^T \left(\mathbf{a} - \tfrac{1}{N}\mathbf{X}^T\mathbf{y}\right) + \lambda_1 \|\mathbf{a}\|_1 + (\mathbf{y} - \mathbf{X}\mathbf{a})^T(\mathbf{y} - \mathbf{X}\mathbf{a}) \to \min(\mathbf{a}) \end{aligned}$$

$$\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = (\hat{a}_{i,\lambda_1,\lambda_2}, i \in I) = \arg\min J_{\text{EN}}(\mathbf{a}|\lambda_1,\lambda_2) \in \mathbb{R}^n,$$

где $\mathbf{a}^* = (1/N)\mathbf{X}^T\mathbf{y}$. Таким образом, можно переписать задачу в следующем виде:

$$\frac{\lambda_1}{1 + \lambda_2/N}\|\mathbf{a}\|_1 + \left[\mathbf{a}^T \frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1 + \lambda_2/N}\mathbf{a} - 2\mathbf{y}^T\mathbf{X}\mathbf{a}\right] \to \min(\mathbf{a}),$$

и есть теорема, которая доказывает что эти задачи эквивалентны.
Следующая идея состоит в том, что мы разбиваем параметры по знаку:

$$\begin{cases} \hat{I}^-_{11}, \lambda_2 = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} < 0\} \\ \hat{I}^0_{1_1,\lambda_2} = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} = 0\}, I = \hat{I}^-_{\lambda_1,\lambda_2} \cup \hat{I}^0_{\lambda_1,\lambda_2} \cup \hat{I}^+_{\lambda_1\lambda_2} \\ \hat{I}^+_{\lambda_1^+,\lambda_2} = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} > 0\} \end{cases}$$

. Дальше идет несколько преобразований с переобозначениями и мы приходим к результату:

$$\begin{aligned} \hat{S}_{\text{LOO}}(\lambda_1,\lambda_2) &= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} \left(\hat{\delta}^{(k)}_{k,\lambda_1,\lambda_2}\right)^2 \\ \hat{\delta}^{(k)}_{k,\lambda_1,\lambda_2} &= y_k - \hat{y}^{(k)}_{k,\lambda_1,\lambda_2} = y_k - \tilde{\mathbf{x}}_k^T \hat{\mathbf{a}}^{(k)}_{\lambda_1,\lambda_2} \end{aligned}$$

и

$$\hat{S}^{\text{EN}}_{\text{LOO}}(\lambda_1,\lambda_2) = \frac{1}{N}\sum_{k=1}^{N} \left(\frac{\hat{\delta}_{k,\lambda_1,\lambda_2} + \frac{1}{N-1}\lambda_2 (y_k q_{k,\lambda_1,\lambda_2} - h_{k,\lambda_1,\lambda_2})}{1 - q_{k,\lambda_1,\lambda_2}} \right)^2 \text{(ElasticNet)}$$

$$\begin{aligned} q_{k,\lambda_1,\lambda_2} &= \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2} \tilde{\mathbf{X}}_{\lambda_1,\lambda_2} + \lambda_2 \tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}} \right)^{-1} \tilde{\mathbf{x}}_k \\ h_{k,\lambda_1,\lambda_2} &= \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2} \tilde{\mathbf{X}}_{\lambda_1,\lambda_2} + \lambda_2 \tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}} \right)^{-1} \tilde{\mathbf{a}}^* \end{aligned}$$

Для практического использования модель должна не только достигать высокого качества предсказаний, но и быть интерпретируемой. Например, исследования показали, что в области медицины предпочитаются модели основанные на деревьях решений, из-за

того, что можно проследить,на основании каких правил строились результаты работы модели. Тем не менее, возможность узнать причины предсказаний модели не являтся основным критерием применимости модели. Если эти причины очень сильно расходятся со здравым смыслом и устоявшимися законами данной области, то наврял ои кто-то решиит прислушаться к такой модели. Эти два качества называются интерпретируемость и правдоподобность модели. В работе [**?**] авторы рассматривают новый способ регуляризации EYE (expert yielded estimates), который включает в себя экспертные знания об отношениях между признаками и зависимой переменной. Авторы расматривают задачу минимизации эмпирического риска

$$\hat{\mathbf{W}} = \arg\min \mathrm{S}(\mathbf{W}, \mathbf{X}, \mathbf{y}) + n\lambda\mathcal{J}(\mathbf{W}, \mathbf{\Gamma})$$

в которой минимизируется сумма функции ошибки и регуляризации $\mathcal{J}$. Имеется множества признаков $\mathcal{D}$, из которых для множества $\mathcal{K} \subseteq \mathcal{D}$ имеется дополнительная информация о том, что эти признаки являются важными в рассматриваемой экспертной области. Следовательно для $\hat{\mathbf{W}}_{\mathcal{D}\setminus\mathcal{K}}$ требуется разреженность, а для $\hat{\mathbf{W}}_{\mathcal{K}}$ - нет. Базовый или «наивный» подход, используемый авторами, заключается в использовании $L1$ и $L2$ регуляризаций, тогда регуляризационный член имеет вид

$$\mathcal{J} = (1-\beta)||\mathbf{\Gamma} \odot \mathbf{W}||_2^2 + \beta||(1-\mathbf{\Gamma}) \odot \mathbf{W}||_1,$$

где параметр β контролирует баланс между признаками из $\mathcal{K}$ и $\mathcal{D} \setminus \mathcal{K}$. Предлагаемое авторами решение имеет следующий вид

$$\mathcal{J} = ||(1-\mathbf{\Gamma}) \odot \mathbf{W}||_1 + \sqrt{||(1-\mathbf{\Gamma}) \odot \mathbf{W}||_1^2 + ||\mathbf{\Gamma} \odot \mathbf{W}||_2^2}, \tag{1}$$

Его использование в задачах стратификации риска пациентов позволило получить модели, в которых основные используемые признаки сильно пересекались с факторами, которые считаются значимыми в медицинской среде. При э том удалось сохранить высокое качество предсказания.

Оценка параметров в иерархической вероятностной модели

В общем случае используется динамическое программирование, как последовательность функций Беллмана. Критерий обучения выглядит здесь

$$\min J(\mathbf{w}_t,\ t = 1, \ldots, T|\mathbf{r},\ c) = \sum_{t=2}^{T}(\mathbf{w}_t - q\mathbf{w}_{t-1})^{\mathsf{T}}\mathbf{D}_r^{-1}(\mathbf{w}_t - q\mathbf{w}_{t-1}) + 2c\sum_{t=1}^{T}\sum_{j=1}^{N_t}\max(0, 1 - y_{j,t}\mathbf{w}_t^{\mathsf{T}}\mathbf{x}_{j,t}) \tag{2}$$

Целевая функция (2) зависит от T параметров $(\mathbf{w}_1, \ldots, \mathbf{w}_T)$, упорядоченных вдоль временной оси. Каждый параметр является $(n+1)$ –мерный вектором $\mathbf{w}_t \in \mathbb{R}^{n+1}$ Критерий

(2) может быть представлен как сумма элементарных функций, каждая из которых зависит от одного или двух векторов (соседних) $\mathbf{w}_t$. Обозначим результат оптимизационной задачи как

$$\widetilde{J}_t(\mathbf{w}_t|\mathbf{r},\ c) = \min_{\mathbf{w}_s, s=1,\ldots,t-1} J_t(\mathbf{w}_s, s=1,\ldots,t|\mathbf{r},c) = \min_{\mathbf{w}_1,\ldots,\mathbf{w}_{t-1}} J_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1},\mathbf{w}_t|\mathbf{r},c)$$

Это функция Беллмана, полностью определенная на обучаемом наборе $\{(\mathbf{X}_t, \mathbf{Y}_t), t = 1,\ldots,T\}$ Главное свойство — реккурентное соотношение

$$\widetilde{J}_0(\mathbf{w}_0|\mathbf{r},\ c) \equiv 0, \mathbf{w}_0 \in \mathbb{R}^{n+1},\ t=0$$

$$\widetilde{J}_t(\mathbf{w}_t|\mathbf{r},\ c) = 2c\sum_{j=1}^{N_t} \max(0, 1 - y_{j,t}\mathbf{w}_t^{\mathsf{T}}\mathbf{x}_{j,t}) +$$

$$+ \min_{\mathbf{w}_{t-1}\in\mathbb{R}^{n+1}} \left[(\mathbf{w}_t - q\mathbf{w}_{t-1})^{\mathsf{T}}\mathbf{D}_r^{-1}(\mathbf{w}_t - q\mathbf{w}_{t-1}) + \widetilde{J}_{t-1}(\mathbf{w}_{t-1}|\mathbf{r},c)\right], \mathbf{w}_t \in \mathbb{R}^{n+1}, t = 1,\ldots,T.$$

Тогда оптимальные параметры для гиперплоскости находятся как $(\hat{\mathbf{w}}_T|\mathbf{r},q,c)$:

$$\hat{\mathbf{w}}_T|\mathbf{r},q,c = \min_{\mathbf{w}_T\in\mathbb{R}^{n+1}} \widetilde{J}_T(\mathbf{w}_T|\mathbf{r},q,c).$$

Для достижения оптимальных скорости вычисления и использовании памяти стоит использовать аппроксимацию функции Беллмана. Тогда оптимизационная задача выпукла и принимает вид

$$\widetilde{J}'_t(\mathbf{w}_t|\mathbf{r},c) = (\mathbf{w}_t - \widetilde{\mathbf{w}}'_t)^{\mathsf{T}}\widetilde{Q}'_t(\mathbf{w}_t - \widetilde{\mathbf{w}}'_t)$$

С условиями

$$\begin{cases} \widetilde{w}'_t = \arg\min \widetilde{J}'_t(\mathbf{w}_t|\mathbf{r},c), \\ \widetilde{Q}'_t = \nabla^2_{\mathbf{w}_t}\widetilde{J}'_t(\mathbf{w}_t|\mathbf{r},c), \qquad \mathbf{w}_t = \widetilde{\mathbf{w}}_t. \end{cases}$$

В случае задачи квадратичного динамического программирования постановка выглядит

$$J(\mathbf{r}|\mathbf{w}_t, t=1,\ldots,T,\mu) = (T-1)\ln|\mathbf{D}_r^{-1}| + \sum_{t=2}^{T}(\mathbf{w}_t - q\mathbf{w}_{t-1})^{\mathsf{T}}\mathbf{D}_r^{-1}(\mathbf{w}_t - q\mathbf{w}_{t-1})$$

$$\min 2\ln G(\mathbf{r}|\mu)$$

Если длина обучаемой последовательности достаточно большая $T \to \infty$, критерий принимает вид

$$J(\mathbf{r}|\mathbf{w}_t, t=1,\ldots,T,\mu) \underset{T\to\infty}{\to} \sum_{i=1}^{n}\left[\left(T-1+\frac{1}{\mu}\right)\ln\frac{1}{r_i} + \frac{1}{r_i}\left(\sum_{t=2}^{T}(\omega_{i,t})^2 + \frac{1}{\mu}\right)\right].$$

В силу того, что сумма здесь выпуклая функция, и производные по $\frac{1}{r_i} = 0$, упрощается итоговая формула для решения $(\hat{\mathbf{r}}|\mathbf{w}_1, \ldots, \mathbf{w}_T, \mu)$:

$$(\hat{r_i}|\mathbf{w}_1, \ldots, \mathbf{w}_T, \mu) = \frac{\sum_{t=2}^{T}(\omega_{i,t})^2 + \frac{1}{\mu}}{T - 1 + (1/\mu)}, \; i = 1, \ldots, n.$$

Регуляризация $L2$ работает с помощью установления баланса между смещением и дисперсией. Но ее недостаток в том, что она не может создать разреженную модель, так как сохраняет все множество признаков. Другой вид регуляризации $L1$ был предложен в [**?**], и он предлагает автоматический выбор переменных. Но он также имеет несколько недостатков, среди которых:

1) Если обозначить за p число независимых переменных, а за n количество объектов в выборке, то в случае $p > n$ lasso регуляризация выбирает максимально n независимых переменных из множества.

2) При наличии групп сильно скоррелированных переменных, lasso регуляризация выбирает только одну переменную из группы, причем не обращая внимания какую именно.

3) В случае $n > p$ и наличии высокой корреляции между переменными было эмпирически показано, что ridge регрессия работает намного лучше lasso.

Таким образом, пукнты 1) и 2) делают lasso неприменимой техникой в некоторых задачах, где требуется отбор признаков. Перечисленные проблемы решаются с помощью другой техники регуляризации elastic net [**?**], которая позволяет производить автоматический отбор переменных, регулировать их веса, а так же выбирать группы коррелярующих признаков. Метод регуляризации elastic net представляет собой добавление в функцию ошибки двух дополнительных слагаемых

$$S(\lambda_1, \lambda_2, \mathbf{w}) = |\mathbf{y} - \mathbf{X}\mathbf{w}|^2 + \lambda_2|\mathbf{w}|^2 + \lambda_1|\mathbf{w}|_1$$

Некоторым обобщением elastic net является регуляризация Support Features Machine (SFM). Задается в виде

$$\min_{\mathbf{w},\mathbf{w_0}} \quad C\sum_{i=1}^{l}(1 - M_i(\mathbf{w}, \mathbf{w_0}))_+ + \sum_{j=1}^{n} R_\mu(w_j),$$

$$R_\mu(w_j) = \begin{cases} 2\mu|w_j|, & |w_j| \le \mu, \\ \mu^2 + w_j^2, & |w_j| \ge \mu. \end{cases}$$

Отбор признаков осуществляется с помощью параметра селективности μ. Также присутствует эффект группировки. Шумовые признаки ($|w_j| < \mu$) подавляются как и в

Lasso, а значимые зависимые признаки группируются также как и в elastic net. На нее похож такой метод как Relevance Features Machine (RFM). Задается в виде

$$\min_{\mathbf{w},\mathbf{w_0}} \quad C\sum_{i=1}^{l}(1 - M_i(\mathbf{w}, \mathbf{w_0}))_+ + \sum_{j=1}^{n}\ln(w_j^2 + \frac{1}{\mu})$$

Здесь отличие от предыдущего метода в том, что тут более совершенный отбор признаков, когда они только совместно обеспечивают хорошее решение.

В общем случае, если функция f — выпуклая, то можно воспользоваться регуляризацией Moreau-Yosida. Записывается в таком виде для $\lambda > 0$

$$M_{\lambda f}(x) = \inf_u(f(u) + \frac{1}{2\lambda}||x - u||_2^2)$$

У такой функции ряд замечательных свойств:

1) $M_{\lambda f}(x)$ — выпуклая функция в силу инфимальной конволюции,

2) Множество точек минимума для f и $M_{\lambda f}(x)$ совпадают,

3) $M_{\lambda f}(x)$ — гладкая функция в силу сильной выпуклости ее первой сопряженной функции и совпадении со второй сопряженной функцие $M_{\lambda f}(x) = M_{\lambda f}^{**}(x)$.

Метапараметры – настраиваемые параметры функции ошибки, используемые для оптимизации параметров модели.

Гиперпарметры – заранее заданные параметры, используемые для оптимизации параметров модели и метапараметров.

Постановка задачи

Мотивация Сети глубокого обучения могут содержать большое количество параметров, таких как веса нейронов. Это позволяет достичь большей точности предсказания, но приводит к увеличению сложности модели, которую мы понимаем как количество и абсолютное значение ее параметров. Увеличение сложности модели приводит к ее нестабильности, то есть сильной зависимости от изменения начальных данных, а так же к увеличению времени работы. В качестве метода борьбы с увеличением сложности при сохранении точности предлагается использовать аддитивную регуляризацию. Требуется исследовать влияние аддитивной регуляризации на точность и сложность модели глубокого обучения.

Наборы данных. Качество предлагаемого подхода к построению функции ошибки оценивается на нескольких реальных наборах данных и одном синтетическом наборе. Выборки взяты из открытого репозитория данных для машинного обучения [?]. Описание всех выборок представлено в табл 1. Синтетический набор данных состоит из признаков с различными свойствами ортогональности и коррелированности друг с другом и к целевой переменной. Процедура генерации синтетических данных описана в работе [1]. Возможны следующие конфигурации синтетических данных.

1. Неполный и скоррелированный: набор данных, содержащий коррелирующие признаки, ортогональные с целевому вектору.

2. Адекватный и случайный: набор данных, содержащий случайные признаки, и имеющий один признак, апроксимирующий целевой вектор.

3. Адекватный и избыточный: набор данных, содержащий признаки, коррелирующие с целевым вектором.

4. Адекватный и скорреллированный: набор данных, содержащий ортогональные признаки, и признаки, коррелирующие с ортогональными. Целевой вектор является суммой ортогональных векторов.

Каждый набор данных разбивается на три части.

1. Обучающая выборка — 60% от исходного набора. На этой выборке модель тренируется, и фиксируются значения параметров.

2. Валидационная выборка — 20% от исходного набора. На этой выборке применяется генетический алгоритм, который ищет оптимальную структуру.

3. Тестовая выборка — 20% от исходного набора. Она никак не участвует в оптимизации структуры модели. Эта выборка используется только для контроля качества — сравнение модели исходной и оптимизированной структуры, а так же сравнение с другими алгоритмами прореживания сетей.

Таблица 1: Описание выборок для экспериментов

Выборка $\mathfrak{D}$	Размер train	Размер val	Размер test	Объекты	Признаки
Credit Card	18000	6000	6000	30000	35
Protein	27438	9146	9146	45730	9
Airbnb	6298	2100	2100	10498	16
Wine quality	2938	980	980	4898	11
Synthetic	1200	400	400	2000	30

Выбор модели Задана выборка

$$(\mathbf{x}_i, y_i), \quad \mathbf{x}_i \in \mathbb{R}^n, \quad y_i \in \mathbb{R}^1, \quad i = 1, \ldots, m, \tag{3}$$

где $\mathbf{x}$ — описание объекта, вектор из n элементов признаков, y — зависимая переменная. Моделью называется отображение $f : (\mathbf{x}, \mathbf{w}) \mapsto y$. Требуется построить аппроксимирующую модель $f(\mathbf{x})$ вида:

$$f = \sigma_k \circ \underset{1\times 1}{\mathbf{w}_k^{\mathsf{T}}}\boldsymbol{\sigma}_{k-1} \circ \mathbf{W}_{k-1}\boldsymbol{\sigma}_{k-2} \circ \cdots \circ \underset{n_2\times 1}{\mathbf{W}_2}\boldsymbol{\sigma}_1 \circ \underset{n_1\times n}{\mathbf{W}_1}\underset{n\times 1}{\mathbf{x}}. \tag{4}$$

Эта модель рассматривается как суперпозиция линейной модели, глубокой нейросети и автоэкнодера. Рассмотрим различные модели как частные случаи (4).
Линейная или логистическая регрессия и один нейрон — имеют вид

$$f(\mathbf{x}, \mathbf{w}) = \boldsymbol{\sigma}(\mathbf{w}^{\mathsf{T}}\mathbf{x}),$$

где $\boldsymbol{\sigma}$ — функция активации, непрерывная монотонная дифференцируемая функция (5), $\mathbf{w}$ — вектор параметров, $\mathbf{x}$ — объект, вектор с присоединенным элементом единица соответствующим аддитивному параметру w_0. При использовании линейной функции активации, получаем линейную регрессию $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}$.

Такую функцию активации мы обозначим $\boldsymbol{\sigma} = \mathbf{id}$. При использовании сигмоидной функции активации, получаем модель логистической регрессии

$$f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x})}. \tag{5}$$

Двухслойная нейронная сеть, состоящая из линейной комбинации нейронов, однослойных нейронных сетей

$$f(\mathbf{x}, \mathbf{w}) = \sigma^{(2)}\left(\sum_{i=1}^{n_2} w_i^{(2)} \cdot \sigma^{(1)}\left(\sum_{j=1}^{n} w_{ij}^{(1)} x_j + w_{i0}^{(1)}\right) + w_0^{(2)}\right) = \sigma \circ \mathbf{w}^{\mathsf{T}}\boldsymbol{\sigma} \circ \mathbf{W}\mathbf{x}.$$

Метод главных компонент. Модель допускает вращения признакового пространства, то есть координаты (признаки) преобразовываются только с помощью поворотов:

$$\mathbf{h} = \mathbf{W}\mathbf{x},$$

где $\mathbf{W}$ — матрица поворота. Она ортогональна:

$$\mathbf{W}\mathbf{W}^{\mathsf{T}} = \mathbf{I}_n. \tag{6}$$

Полученное пространство образов $\mathbf{h}$ называется скрытым. Происходит преобразование без потерь.

При удалении нескольких строк матрицы $\mathbf{W}$, например их число $u < n$, полученный вектор $\mathbf{h}$ имеет размер $u \times 1$. Получается проекция $\mathbf{h}$ вектора $\mathbf{x}$. Согласно теореме Рао С.Р. [**?**], первые u главных компонент восстанавливают $\mathbf{h}$ оптимальным способом,

$$\mathbf{r}(\mathbf{x}) = \mathbf{W}^{\mathsf{T}}\mathbf{h}.$$

Автокодировщик $\mathbf{h}$ — это монотонное нелинейное отображение входного вектора свободных переменных $\mathbf{x} \in \mathbb{R}^n$ в скрытое представление $\mathbf{h} \in \mathbb{R}^u$ вида:

$$\mathbf{h}(\mathbf{x}) = \boldsymbol{\sigma}(\underset{u\times n}{\mathbf{W}}\mathbf{x} + \mathbf{b}).$$

В случае $\boldsymbol{\sigma} = \mathbf{id}$ и (6) автокодировщик тождественен методу главных компонент. Скрытое представление $\mathbf{h}$ реконструирует вектор $\mathbf{x}$ линейно:

$$\mathbf{r}(\mathbf{x}) = \underset{n\times u}{\mathbf{W}'}\mathbf{h} + \mathbf{w}_0'.$$

Функция ошибки и критерии качества модели Ключевой идеей данной работы является построение новой функции ошибки с использованием метапараметров аддитивной регуляризации.

Предлагается использовать композитную функцию ошибки. Она состоит из нескольких слагаемых. Первое слагаемое соответствует точности восстановления зависимой переменной. Второе слагаемое это точность реконструкции независимой переменной автокодировщиком. Остальные r слагаемых отвечают за аддитивную регуляризацию. Задача (10) является задачей минимизации функции L, включающее слагаемое (8) и (9) для оптимизации параметров модели (4)

$$L = E_x + \mathcal{S} + \lambda_1 S_1 + \cdots + \lambda_k S_k = E_x + \mathcal{S} + \sum_{i=1}^{r} \lambda_i^T \mathbb{S}_i(\mathbf{W}_i). \tag{7}$$

Требуется создать каталог слагаемых функции ошибки.

Тип регуляризатора/слагаемого	Роль
$\|\|\mathbf{y} - f(\mathbf{W})\|\|_2^2$,	Ошибка выхода нейронной сети
$\|\|\mathbf{x} - \mathbf{r}(\mathbf{x})\|\|_2^2$	Ошибка восстановления на каждом слое
$\|\|\mathbf{w} - \mathbf{w}_0\|\|_1$, $\|\|\mathbf{w} - \mathbf{w}_0\|\|_2^2$,	L_1 и L_2 регуляризация
$\|\|\mathbf{W} - \mathbf{I}\|\|$	Штраф за различие матрицы одного слоя от тождественного преобразования
$\|\|\mathbf{W}\mathbf{W}^T - \mathbf{I}\|\|$	Штраф за различие матрицы одного слоя от метода главных компонент
$\|\mathbf{T}W\|$	Тихоновская регуляризация

Таблица 2: Каталог слагаемых функции ошибки

Требуется создать расписание оптимизации параметров регуляризации λ. Требуется назначать лямбду в зависимости от номера итерации. Если оптимизация сети только началась, то важно подготовить выборку, чтобы на последнем слое нейросети она была простой. Поэтому требуется, чтобы начальные автоекнодеры работали хорошо. Для выбора λ предполагается использовать эмпирический подход и исследовать различные

техники, как например подходы для выбора шага обучения. Лямбда – точка парето оптимального фронта.

В работе –(прошлая статья)– поиск структуры работал через добавление или удаление элементов из структуры. Это решение по сути является лихорадочным метанием, которое работает как полный перебор. Требуется предложить стратегию направленного поиска. Такая стратегия, конечно, работает чуть хуже чем полный перебор. Алгоритм градиентного спуска сходится довольно быстро. Требуется исследовать возможность поиска оптимальной структуры через сложность.

В работе [**?**] представлена таблица свойств различных регуляризаций. В работе рассматриваются нормы со степенью один и два, то есть метрики $L1$ и $L2$. Требуется исследовать свойства метрика с различными значениями степени.

Используется три вида критериев качества: точность, устойчивость и сложность.

Точность. Когда в качестве используемой модели выступает нейросеть или линейная регрессия, то функция ошибки имеет вид:

$$\mathcal{S} = \sum_{i \in \mathcal{I}} \big(y_i - f(\mathbf{x}_i)\big)^2. \tag{8}$$

Эта функция ошибки включает в себя полученные предсказания модели и значения зависимых переменных. В задачах регрессии точность аппроксимации имеет вид:

$$\text{MAE} = \frac{\sum\limits_{i=1}^{m} |y_i - f(\mathbf{x}_i)|}{m}.$$

При включении в модель (4) метода главных компонент или автокодировщика, метки объектов не используются. Функция ошибки штрафует невязки восстановленного объекта:

$$E_{\mathbf{x}} = \sum_{i \in \mathcal{I}} \|\mathbf{x}_i - \mathbf{r}(\mathbf{x}_i)\|_2^2, \tag{9}$$

где $\mathbf{r}(\mathbf{x})$ это линейная реконструкция объекта $\mathbf{x}$. Параметры автокодировщика

$$\mathbf{W}_{\text{AE}} = \{\mathbf{W}', \mathbf{W}, \mathbf{b}', \mathbf{b}\}$$

оптимизированы таким образом (9), чтобы приблизить реконструкцию $\mathbf{r}(\mathbf{x})$ к исходному вектору $\mathbf{x}$.

Процедура оптимизации параметров композитной функции (7):

1) оптимизируются параметры автокодировщика согласно (9),

2) заданные параметры фиксируются,

3) настраиваются метапараметры λ.

4) оптимизируются параметры модели согласно (8).

Сложность. Введем отношение порядка $\succ$ на множестве значений сложности. Это отношение задается множеством параметров модели:

1) один параметр: $w \in \mathbb{R}^1 \succ w \in \lambda_1[0,1] + \lambda_0 \succ w \in c + \lambda_0$,

2) вектор(нейрон): $\mathbf{w} \in \mathbb{R}^n \succ \|\mathbf{w}\|^2 = 1 \succ \mathbf{w} = \text{const}$,

3) матрица(слой): $\mathbf{W} \in \mathbb{R}^{c \times n} \succ \mathbf{W}^\intercal\mathbf{W} = \mathbf{I} \succ \mathbf{W} = \text{const}$.

Множество, которому принадлежит сложность модели – порядковое. Исходя из введенного понятия сложности модели упорядочены по возрастанию сложности:

1) линейная регрессия, $\sigma^{'} = \text{id}, \sigma = \text{id}, \mathbf{W} = \mathbf{I}_n$,

2) линейная регрессия и метод главных компонент, $\sigma^{'} = \text{id}, \mathbf{W}^\intercal\mathbf{W} = \mathbf{I}_n$,

3) линейная модель и автокодировщик, $\mathbf{W}^\intercal\mathbf{W} \neq \mathbf{I}_n$,

4) линейная модель и стэк автокодировщиков, представимый в виде суперпозиции (4),

5) двухслойная нейронная сеть,

6) глубокая нейронная сеть.

Аддитивная регуляризация Рассматриваются регуляризации следующих типов:

1. Lasso или L_1 регуляризация вида:

$$S_1(w) = \lambda_1\|w\|_1$$

2. Штраф за количество слоев в нейронной сети:

$$S_2(k) = \lambda_2 \cdot k$$

3. Штраф за неортогональность матрицы:

$$S_3(W) = \lambda_3\|WW^T - I\|$$

4. https://towardsdatascience.com/tikhonov-regularization-an-example-other-than-l2-8922ba51253dHe видов Тихоновской регуляризации:

   4.1. Ridge или L_2 регуляризация вида:

$$S_4(w) = \lambda_4\|w\|_2^2$$

4.2. Штраф за частоту появления весов

$$A = \frac{1}{3}\begin{bmatrix} \frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{2}{3} \end{bmatrix}$$

$$S_5(W) = \lambda_5 \|(I - A)W\|$$

4.3. Штраф за локальную разницу в весах

$$B = \begin{bmatrix} -2 & 2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 2 \end{bmatrix}$$

$$S_6(W) = \lambda_6 \|BW\|$$

Устойчивость — это минимум дисперсии функции ошибки (3):

$$\mathsf{D}(S) \to \min .$$

Формулировка задачи Таким образом, задача сводится к следующему виду:

$$L(\mathbf{w}, \lambda) = \|y - f(\mathbf{w}, x)\|_2^2 + E_x + \sum_{i=1}^{r} \lambda_i^T \mathbb{S}_i(\mathbf{W}_i)$$

$$\mathbf{w} = \arg\min L(f|\lambda) \tag{10}$$

your model in the class of models, restrictions on the class of models, the error function (and its inference) or a loss function, or a quality criterion, cross-validation procedure, restrictions to the solutions, external (industrial) quality criteria, the optimization statement as argmin.

Список литературы

[1] AM Katrutsa and VV Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.