

Метрический анализ пространства параметров глубоких нейросетей

Эрнест Р. Насыров

`nasyrov.rr@phystech.edu`

Исследуется проблема снижения размерности пространства параметров модели машинного обучения. Решается задача восстановления временного ряда. Для восстановления используются авторегрессионные модели: линейные, автоэнкодеры, рекуррентные сети — с непрерывным и дискретным временем. Проводится метрический анализ пространства параметров модели. Предполагается, что отдельные параметры модели, случайные величины, собираются в векторы, многомерные случайные величины, анализ взаимного расположения которых в пространстве и представляет предмет исследования данной работы. Этот анализ снижает число параметров модели, оценивает значимости параметров, отбирая их. Для определения положения вектора параметров в пространстве оцениваются его математическое ожидание и матрица ковариации с помощью методов *бутстрэп* и *вариационного вывода*. Эксперименты проводятся на задачах восстановления синтетических временных рядов, квазипериодических показаний акселерометра, периодических видеоданных. Для восстановления применяются модели SSA (singular spectrum analysis), нелинейного PCA (principal component analysis), RNN (recurrent neural network), Neural ODE (neural ordinary differential equations).

Ключевые слова: *Временные ряды; снижение размерности; релевантность параметров; пространство параметров; выбор модели.*

1 Introduction

Ключевые слова: временные ряды, снижение размерности, релевантность параметров, пространство параметров, выбор модели.

Высокоразмерные данные избыточны, что представляет сложность для их эффективной обработки и использования. В работе решается задача снижения размерности признакового

описания объекта. Ее базовый принцип состоит в том, чтобы отобразить высокоразмерное признаковое пространство в низкоразмерное, сохраняя важную информацию о данных [Jia et al., 2022].

На текущий момент известно много методов снижения размерности данных. В работе [Örnek and Vural, 2019] снижения размерности достигается за счет построения дифференцируемой функции эмбединга в низкоразмерное представление, а в [Cunningham and Yu, 2014] обсуждаются линейные методы. В работе [Isachenko and Strijov, 2022] задача снижения размерности решается для предсказания движения конечностей человека по электрокортикограмме с использованием метода QPFS (quadratic programming feature selection), учитывающем мультикоррелированность и входных, и целевых признаков.

Наряду с задачей снижения размерности входных данных стоит задача выбора оптимальной структуры модели. В случае оптимизации структуры нейросети, большое внимание уделено изучению признакового пространства модели. В работах [Hassibi et al., 1993] и [Dong et al., 2017] применяется метод OBS (Optimal Brain Surgeon), состоящий в удалении весов сети с сохранением ее качества аппроксимации, причем выбор удаляемых весов производится с помощью вычисления гессиана функции ошибки по весам.

В статье [Грабовой et al., 2019] приводится метод первого порядка, решающий задачу удаления весов, основанный на нахождении дисперсии градиента функции ошибки по параметру и анализе ковариационной матрицы параметров, а в статье Грабовой et al. [2020] нерелевантные веса не удаляют, а прекращают их обучение.

Приведенные выше задачи снижения размерности данных и выбора оптимальной структуры нейросети основаны на исследовании пространства входных данных и пространства признаков соответственно. Существенный недостаток предыдущих работ состоит в том, что в них анализируются *отдельные* параметры (скаляры) моделей и их взаимозависимость. Тем самым не учитывается, что на входные данные действуют *вектора* параметров

31 посредством скалярных произведений, то есть упускается из виду простая *структура*
 32 преобразования.

33 В данной работе решается задача восстановления временного ряда, в рамках которой
 34 исследуем проблему снижения размерности пространства параметров модели. Снижение
 35 размерности основано на анализе сопряженного пространства ко входному. Оно связывает
 36 входное пространство и пространство параметров.

37 Данное исследование в большой степени полагается на простоту устройства глубоких
 38 нейросетей, которые являются композицией линейных и простых нелинейных функций
 39 (функций активации). Составной блок нейросети описан формой:

$$40 \quad \mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad \mathbf{y}, \mathbf{b} \in \mathbb{R}^m, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{W} \in \mathbb{R}^{m \times n}, \quad \sigma : \mathbb{R} \rightarrow \mathbb{R}.$$

41 Раньше элементы \mathbf{W}_{ij} исследовались по-отдельности, как скаляры. Авторы работы пред-
 42 лагают изучать их как векторы-строки

$$43 \quad \mathbf{w}_1, \dots, \mathbf{w}_m : \mathbf{W} = \begin{pmatrix} \mathbf{w}_1^\top \\ \dots \\ \mathbf{w}_m^\top \end{pmatrix}.$$

44 В нейросети эти строки обычно называются *нейронами*. В SSA $\sigma = Id$, а матрица $\mathbf{W} = \mathbf{W}_k$
 45 это приближение истинной матрицы фазовых траекторий \mathbf{X} (матрицы Ганкеля) суммой k
 46 элементарных матриц.

47 Обозначим $\mathbf{x} = [x_1, \dots, x_N]^\top$, $x_i \in \mathbb{R}$ — временной ряд, $1 \leq n \leq N$ — ширина окна. Точка
 48 $\mathbf{x}_t = [x_t, \dots, x_{t+n-1}]^\top$ является точкой фазовой траектории временного ряда в траекторном
 49 пространстве $\mathbb{H}_{\mathbf{x}} \subset \mathbb{R}^n$.

50 Предполагается, что каждая точка фазовой траектории распределена нормально вокруг
 51 своего матожидания. Тогда и временной ряд является случайным, поэтому результат
 52 обучения модели на нем, то есть параметры обученной модели, будут случайными.

53 В работе исследуется положение случайных векторов параметров модели \mathbf{w}_i в метриче-
 54 ском пространстве. С помощью методов бутстрэпа и вариационного вывода [Hastie et al.,

2009] оцениваются их матожидания $\mathbf{e}_i = \mathbf{E}(\mathbf{w}_i)$ и ковариационные матрицы $\mathbf{D}(\mathbf{w}_i) = \mathbf{A}_i^{-1}$. Мы работаем в гипотезе, что эти векторы \mathbf{w}_i распределены нормально, таким образом пара $(\mathbf{e}_i, \mathbf{A}_i^{-1})$ полностью описывает вероятностное распределение вектора \mathbf{w}_i .

В качестве графического анализа пространства изображаются положения этих векторов как смеси гауссианов. На рис. 1 изображены плотности функции распределения трех гауссовских векторов (вертикальная ось) в зависимости от их положения на плоскости. В каждой точке плотность равна сумме плотностей трех распределений, отнормированная таким образом, чтобы площадь под графиком равнялась 1.

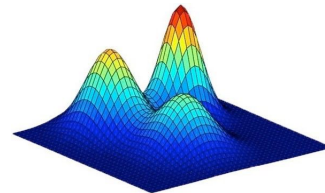
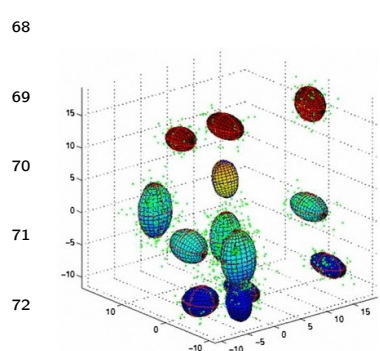


Рис. 1 Смесь гауссианов трех 2-х мерных векторов.

Точкам максимума куполов распределения соответствуют матожидания векторов, а их форма определяется матрицей ковариации \mathbf{A} (добавить стиль). Таким образом, чем ниже и шире 'купол', тем больше дисперсия и наоборот.



На рис. 2 изображены эллипсы, соответствующие 95% доверительным областям для гауссовских векторов в 3-х мерном пространстве. Чем больше ширина эллипса вдоль направления, тем больше дисперсия вектора по этому направлению.

Уменьшение размерности достигается за счет метрического анализа пространства векторов-параметров путем отбора релевантных строк (с малой дисперсией), замены мультикоррелирующих строк на их линейную композицию с помощью обобщения алгоритма QPFS, изучения структуры сообществ строк.

В качестве базовых моделей используются SSA ([Golyandina et al., 2001]), нелинейный PCA, RNN ([Bronstein et al., 2021]), VAE ([Kingma et al., 2019]) и Neural ODE ([Chen et al., 2018]).

Задача восстановления временного ряда решается на синтетических данных зашумленного \sin , данных показания акселерометра в датасете MotionSense3 [Malekzadeh et al., 2018], периодичных видеоданных.

2 Problem statement

Пусть имеется множество из m временных рядов $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_m\}$, $\mathbf{s}_i = [\mathbf{s}_i^1, \dots, \mathbf{s}_i^T]$, $\mathbf{s}_i^j \in \mathbb{R}$, где n — длина сигналов. Каждый временной ряд — последовательность измерений величины в течение времени.

Definition 1. Временное представление $\mathbf{x}_t = [\mathbf{s}_1^t, \dots, \mathbf{s}_m^t]^\top \in \mathbb{R}^m$ состоит из измерений временных рядов в момент времени t .

Definition 2. Предыстория длины h для момента времени t множества временных рядов \mathcal{S} — это матрица $\mathbf{X}_{t,h} = [\mathbf{x}_{t-h+1}, \dots, \mathbf{x}_t]^\top \in \mathbb{R}^{h \times m}$.

Definition 3. Горизонт прогнозирования длины p для момента времени t множества временных рядов \mathcal{S} — это матрица $\mathbf{Y}_{t,p} = [\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+p}]^\top$.

Definition 4. Прогностическая модель $\mathbf{f}^{\text{AR}} : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{p \times m}$ является авторегрессионной моделью, которая по предыстории $\mathbf{X}_{t,h}$ предсказывает горизонт планирования $\mathbf{Y}_{t,p}$.

Решается задача авторегрессионного декодирования. Она состоит в построении прогностической модели \mathbf{f}^{AR} , дающий горизонт прогнозирования множества временных рядов по предыстории того же множества рядов. В дальнейшем будем считать, что восстанавливаем 1 временной ряд, то есть что $m = 1$.

Обозначим множество всех одномерных временных рядов через \mathbb{S} :

$$\mathbb{S} = \bigcup_{n=1}^{+\infty} \{[s_1, \dots, s_n] \in \mathbb{R}^n\}.$$

Тогда прогностическая модель это функция $f^{\text{AR}} : \mathbb{S} \rightarrow \mathbb{R}$. Изначальный временной ряд $\mathbf{s} = [s_1, \dots, s_T]$ делится на две части $\mathbf{s} = [\mathbf{s}^H | \mathbf{s}^T]$, $\mathbf{s}^H = [s_1, \dots, s_h]$, $\mathbf{s}^T = [s_{h+1}, \dots, s_T]$. Задача

104 состоит в том, чтобы предсказать \mathbf{s}^T с максимальной точностью. Предсказание происходит
105 следующим образом:

- 106 1. С помощью модели f предсказывается \hat{s}_{h+1} .
- 107 2. Предсказанный элемент \hat{s}_{h+1} вместе с исходным временным рядом \mathbf{s}^H подаются на
108 вход f для предсказания \hat{s}_{h+2} .
- 109 3. Шаги 1 – 2 повторяются, пока не будет предсказан весь $\hat{\mathbf{s}}^T = [\hat{s}_{h+1}, \dots, \hat{s}_T]$.

110 Для упрощения нотации обозначим $f(\mathbf{s}^H) = \hat{\mathbf{s}}^T = [\hat{s}_{t+1}, \dots, \hat{s}_T]$.

Модель $f = f(\mathbf{w}, \mathbf{s})$, $\mathbf{w} \in \mathbb{W}$, $\mathbf{s} = [s_1, \dots, s_t] \in \mathbb{R}^t$ выбирается из некоего параметрического семейства. Параметры модели выбираются таким образом, чтобы минимизировать функцию ошибки $S = S(\mathbf{w}|\mathbf{s}, f)$:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{W}} S(\mathbf{w}|\mathbf{s}, f).$$

В работе будет использоваться функция ошибки MSE, то есть

$$S(\mathbf{w}|\mathbf{s}, f) = \sum_{t=h+1}^T (s_t - \hat{s}_t)^2.$$

111 .

112 В качестве входных данных модели получают матрицу фазовых траекторий. Она
113 строится по временному ряду $\mathbf{s} = [s_1, \dots, s_T]$ и ширине окна h следующим образом:

$$114 \quad \mathbf{X} = \begin{pmatrix} s_1 & \dots & s_k \\ s_2 & \dots & s_{k+1} \\ \dots & \dots & \dots \\ s_h & \dots & s_T \end{pmatrix} = [X_1 : \dots : X_k], k = T - h + 1, X_i = [s_i, s_{i+1}, \dots, s_{i+h-1}] \in \mathbb{R}^h.$$

115 Траекторная матрица является матрицей Ганкеля, так как каждая диагональ вида
116 $i + j = \text{const}$ содержит одинаковые элементы. Векторы X_i являются точками фазовой
117 траектории сигнала.

118 В качестве альтернативных моделей для восстановления временного ряда будем исполь-
119 зовать 1) SSA, 2) двуслойную нейросеть с ортогональными линейными преобразованиями
120 (нелинейный PCA), 3) RNN и 4) Neural ODE. Остановимся на каждой из них подробнее.

2.1 SSA

В модели SSA восстановление временного ряда получается за счет разложения матрицы фазовой траектории в сумму одноранговых матриц.

Далее вычисляются собственные значения и соответствующие собственным подпространствам ортонормированные системы векторов матрицы

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T \in \mathbb{R}^{h \times h}.$$

Обозначим через

$$\lambda_1 \geq \dots \geq \lambda_L \geq 0$$

собственные значения \mathbf{S} ,

$$u_1, \dots, u_h$$

ортонормированную систему собственных векторов, соответствующую собственным значениям,

$$v_i = \frac{\mathbf{X}^T u_i}{\sqrt{\lambda_i}} \quad (i = 1, \dots, h).$$

Тогда

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d, \quad \mathbf{X}_i = \sqrt{\lambda_i} u_i v_i^T$$

— SVD-разложение \mathbf{X} . Далее происходит группировка матриц \mathbf{X}_i , их ганкелизация и восстанавливается матрица сигнала

$$\hat{\mathbf{X}} = \tilde{\mathbf{X}}_{I_1} + \dots + \tilde{\mathbf{X}}_{I_m}, \quad I_1 \sqcup \dots \sqcup I_m \subset \{1, \dots, h\}, \quad \tilde{\mathbf{X}}_I = \text{hank}\left(\sum_{i \in I} \mathbf{X}_i\right).$$

Где $\text{hank}(X)$ — ганкелизация матрицы X , состоящая в том, что на каждой диагонали вида $i + j = \text{const}$ все элементы заменяются на их среднее арифметическое. Параметрами модели SSA являются множества I_1, \dots, I_m .

2.2 Нейросеть

Модель двуслойной нейросети с ортогональными преобразованиями задается следующим образом:

$$\mathbf{f}(\mathbf{x}) = \sigma(\mathbf{W}_1^T \cdot \sigma(\mathbf{W}_2^T \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2),$$

146 где

$$147 \quad \mathbf{x} \in \mathbb{R}^h, \mathbf{W}_2 \in \mathbb{R}^{h \times d}, \mathbf{W}_1 \in \mathbb{R}^{d \times h}, b_1 \in \mathbb{R}^h, b_2 \in \mathbb{R}^h; \mathbf{W}_1^T \mathbf{W}_1 = \mathbf{I}, \mathbf{W}_2 \mathbf{W}_2^T = \mathbf{I}.$$

148 Последние два условия гарантируют, что преобразования будет ортогональным. Здесь
149 нейросеть восстанавливает значения сигнала длиной в h моментов времени.

150 2.3 RNN

151 Модель RNN задается следующим образом:

$$152 \quad \mathbf{h}_t = \sigma(\mathbf{W} \cdot \mathbf{h}_{t-1} + \mathbf{V} \cdot \mathbf{x}_t),$$

153

$$154 \quad \mathbf{s}_{t+1} = \tanh(\mathbf{w}_o^T \cdot \mathbf{h}_t),$$

155 где $\mathbf{h}_t \in \mathbb{R}^d$ — скрытое состояние RNN в момент времени t , $\mathbf{W} \in \mathbb{R}^{d \times d}$, $\mathbf{V} \in \mathbb{R}^{d \times h}$.

156 $\mathbf{x}_t = [s_{t-h+1}, \dots, s_t] \in \mathbb{R}^h$ - временной ряд, подающийся на вход модели, $\mathbf{w}_o \in \mathbb{R}^d$, $s_{t+1} \in \mathbb{R}$ —
157 прогноз значения сигнала в момент времени $t + 1$. Параметрами модели являются матрицы
158 \mathbf{W} , \mathbf{V} , \mathbf{w}_o а также начальное скрытое состояние \mathbf{h}_0 , которое мы зафиксируем $\mathbf{h}_0 = 0$.

159 О входном временном ряде выдвинута гипотеза о том, что точки на фазовой траектории
160 распределены по нормальному закону, то есть что $\mathbf{x}_t = [s_t, \dots, s_{t+h-1}] \sim \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{B})$, где
161 $\hat{\mathbf{x}}_t$ — это матожидание точки фазовой траектории в момент времени t , а \mathbf{B} — матрица
162 ковариации. Мы также предполагаем, что [ссылаться на формулу, где впервые введена
163 $\mathbf{W}] \mathbf{w} \sim \mathcal{N}(\hat{\mathbf{w}}, \mathbf{A})$.

164 В работе оцениваются матожидание вектора параметров модели $\hat{\mathbf{w}}$, а также матрица
165 ковариации параметров \mathbf{A} с помощью методов бутстрепа и вариационного вывода.

166 Для метрического анализа пространства параметров выбираются набор множеств ин-
167 дексов $\mathcal{I}_1 \sqcup \dots \sqcup \mathcal{I}_k \subset \{1, \dots, \text{len}(\mathbf{w})\}$, рассматриваются соответствующие им подвектора
168 параметров $\mathbf{w}_{\mathcal{I}_1}, \dots, \mathbf{w}_{\mathcal{I}_k}$. Считается, что каждый $\mathbf{w}_{\mathcal{I}_j}$ соответствует «смысловой едини-
169 це» модели. В нейросетевых моделях это строки матрицы линейного преобразования
170 пространства параметров \mathbf{W} , которые обычно называют *нейронами*.

Для каждого \mathbf{w}_T оценивается матожидание $\hat{\mathbf{w}}_T$ и ковариационная матрица \mathbf{A}_T , с помощью которых и проводится метрический анализ пространства параметров.

3 Computational experiment

3.1 SSA

Скалярный временной ряд $\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon$ состоит из $n = 500$ значений зашумленного синуса, измеренного в точках интервала $[-\pi, \pi]$ с равномерным шагом. Использован шум из распределения $\varepsilon \sim \mathcal{N}(0, 0.2)$. Изображение данных представлено на рис. 3.

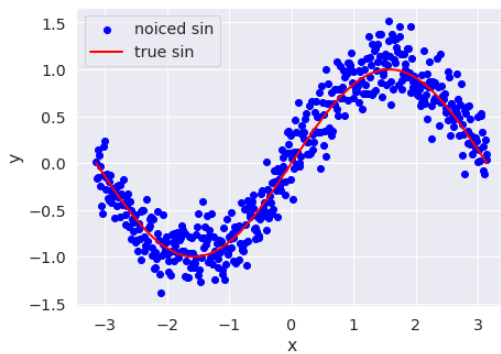
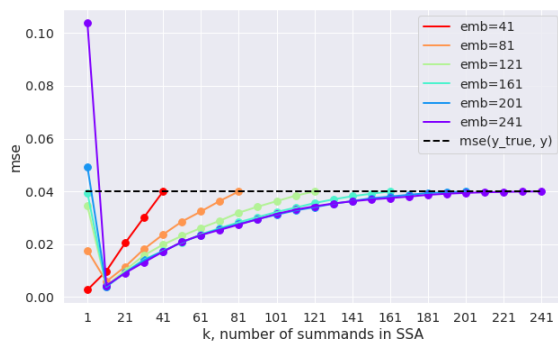


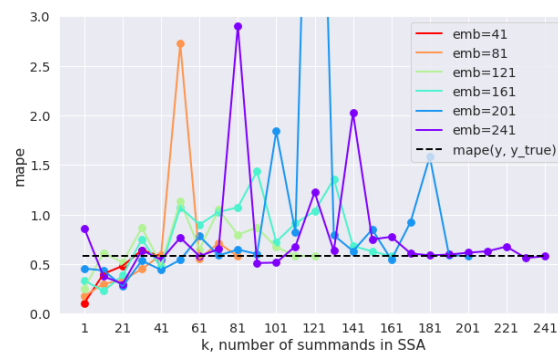
Рис. 3 Зашумленный синус.

С помощью метода SSA построена зависимость точности восстановления временного ряда \mathbf{x} из шумного ряда $\tilde{\mathbf{x}}$ в зависимости от ширины окна h и количества главных компонент k при восстановлении матрицы \mathbf{X} .

В качестве критерия качества использовались MSE и MAPE, замеренные между восстановленным рядом $\hat{\mathbf{x}}$ и истинным значением ряда \mathbf{x} . Результаты представлены на рис. ?? и рис. ??.



(а): MSE для различных значений значения h (emb_size) и k . Черной линией выделен MSE между чистым и зашумленным рядом.



(б): MAPE для различных значений значения h (emb_size) и k . Черной линией выделен MAPE между чистым и зашумленным рядом.

Различная длина графиков обусловлена тем, что длина окна h не может быть больше чем количество слагаемых в SVD-восстановлении ряда. Видно, что истинное значение ряда с большой точностью восстанавливается при достаточно небольшом количестве слагаемых (< 20), что соответствует точке минимума всех графиков на левом рисунке. Чем больше компонент используется, тем больше становится ошибка и при количестве компонент равном длине окна MSE становится таким же, как и MSE между истинным значением ряда и его шумной версией (графики приближаются к горизонтальной пунктирной линии).

Полученные результаты подтверждают практическую состоятельность SSA: алгоритм восстанавливает главный тренд при небольшом количестве сингулярных слагаемых \mathbf{X}_i .

3.2 Двуслойная нейросеть с ортогональными преобразованиями

Скалярный временной ряд $\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon$ состоит из $n = 500$ значений зашумленного синуса, измеренного в точках интервала $[-4\pi, 4\pi]$ с равномерным шагом. Использован шум из распределения $\varepsilon \sim \mathcal{N}(0, 0.2)$. Изображение данных представлено на рис. 4.

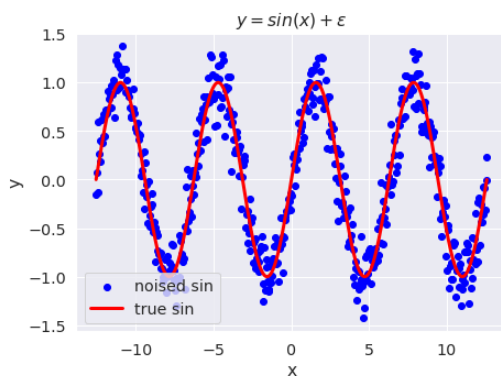


Рис. 4 Зашумленный синус.

С помощью нейросети решалась задача восстановления временного ряда. Скрытая размерность нейросети $d = 3$, размерность траекторного пространства $h = 20$.

На случайных подвыборках обучающей выборки были обучены $N = 100$ нейросетей на 25 эпохах. Оценены матожидания и матрицы ковариаций столбцов параметра $\mathbf{W}_1 \in \mathbf{R}^{3 \times 20}$. Полученные матрицы визуализированы в 3-х мерном

пространстве рис 5.

210 Все параметры-нейроны на рис. ? имеют форму
211 вытянутых эллипсов, расположенных на удалении
212 друг от друга. Часть эллипсов вытянута в одну и
213 ту же сторону (уменьшения оси OY). В целом все
214 эти нейроны кажутся довольно независимыми и не
215 имеющими выраженной структуры сообществ.

216 В следующем эксперименте с нейросетью бра-
217 лась архитектура $10 - 3 - 20$, которая означает, что
218 на вход подаются $h_1 = 10$ -мерные векторы, размер-
219 ность скрытого пространства $d = 3$, а восстановить
220 нужно $h_1 = 20$ моментов времени.

221 На случайных подвыборках обучающей выборки
222 были обучены $N = 100$ нейросетей на 70 эпохах. Оценены матожидания и матрицы
223 ковариаций столбцов параметра $\mathbf{W}_1 \in \mathbf{R}^{3 \times 10}$. Полученные матрицы визуализированы в 3-х
224 мерном пространстве рис 6.

225 Здесь, в отличие от предыдущего эксперимента наблюдается разделение на 3 сообщества,
226 соответствующие оранжевому, красному и фиолетовому эллипсоиду, которые имеют самую
227 большую дисперсию среди нейронов (так как они большего объема).

228 Это частично подтверждает гипотезу о существовании структуры сообществ у нейронов.

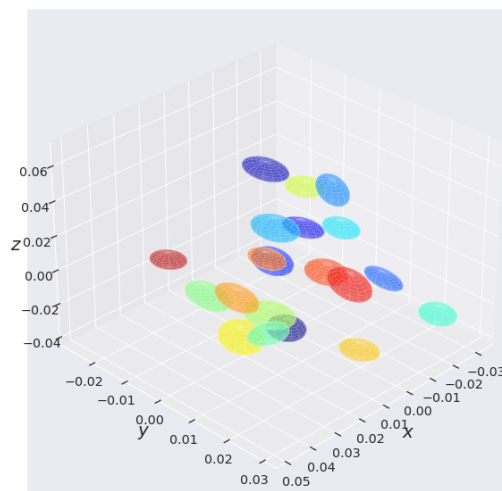


Рис. 5 Доверительные области 3-х мерных векторов двуслойной нейросети архитектуры 20-3-20.

4 Решение задачи

(заимствовано из диссертации Грабового). Решается задача байесовской дистилляции модели учителя в байесовскую версию модели ученика.

В качестве модели учителя используется модель HTNet [Peterson et al., 2021], которая основывается на EEGNet [Lawhern et al., 2018]. EEGNet является сверточной неросетью с 5 последовательными слоями: temporal convolution (различная дискретизация сигнала для получения фильтров *частоты*), depthwise convolution (сверточные фильтры для получения частото-специфичных признаков), separable convolution (depthwise convolution для суммаризации информации внутри каждой частоты и затем pointwise convolution для смешения признаков).

Решается задача предсказания движения рук по электрокортикограмме головного мозга.

Задана выборка

$$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{y}_i)\}_{i=1}^N, \mathbf{s}_i \in \mathbb{R}^n, \mathbf{y}_i \in \mathbb{R}^m,$$

где \mathbf{s}_i — i -ый временной ряд электрокортикограммы, \mathbf{y}_i — соответствующий временной ряд движения руки.

Задана модель учителя в виде суперпозиций линейных и нелинейных преобразований:

$$\mathbf{f} = \sigma \circ \mathbf{U}_T \circ \sigma \circ \mathbf{U}_{T-1} \circ \dots \circ \mathbf{U}_2 \circ \sigma \circ \mathbf{U}_1,$$

где T — число слоев модели учителя (от 5 до 8 у применяемых далее моделей), σ — функция активации, \mathbf{U}_t — матрица линейного преобразования. Матрицы \mathbf{U}_t , параметры модели

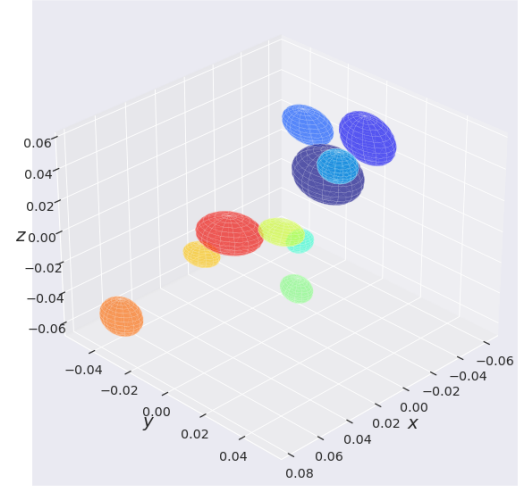


Рис. 6 Доверительные области 3-х мерных векторов двуслойной нейросети архитектуры 10-3-20.

253 учителя, соединяются в вектор параметров \mathbf{u} модели учителя \mathbf{f} :

254
$$\mathbf{u} = \text{vec}([\mathbf{U}_T, \mathbf{U}_{T-1}, \dots, \mathbf{U}_1]).$$

255 Мы работаем в предположении, что параметры учителя распределены нормально. Для
256 модели учителя оценивается апостериорное распределение вектора параметров

257
$$p(\mathbf{u}) = p(\mathcal{N}(\bar{\mathbf{u}}_{ps}, \mathbf{A}_{ps}^{-1})).$$

258 В качестве модели ученика используется байесовская нейронная сеть [вставить ссылку на
259 статью], задаваемая аналогично модели учителя:

260
$$\mathbf{g} = \sigma \circ \mathbf{W}_L \circ \sigma \circ \mathbf{W}_{L-1} \circ \dots \circ \mathbf{W}_2 \circ \sigma \circ \mathbf{W}_1.$$

261 Мы считаем, что

262
$$\mathbf{w} \sim \mathbf{N}(\bar{\mathbf{w}}_t, \mathbf{B}^{-1}).$$

263 Алгоритм римановской дистилляции представлен ниже.

Алгоритм 1 Алгоритм Римановской дистилляции

1. Зафиксировать структуру модели ученика \mathbf{g} и учителя \mathbf{g} , то есть параметры T, L и размеры матриц $\mathbf{U}_t, \mathbf{W}_t$.
 2. Обучить модель учителя \mathbf{g} .
 3. Оценить апостериорное распределение параметров учителя $p_{ps}(\mathbf{u})$, то есть параметры $\bar{\mathbf{u}}, \mathbf{A}^{-1}$.
 4. Назначить параметрам ученика априорное распределение $p(\mathbf{w}) \sim p_{ps}(\mathbf{u})$. Где \sim обозначает преобразование распределения, описанное в работе [Grabovoy and Strijov, 2021].
 5. Обучить байесовскую сеть ученика на ответах учителя, получив апостериорное распределение параметров ученика $p(\mathbf{w}|\mathcal{D}) = p(\mathbf{N}(\bar{\mathbf{w}}_t, \mathbf{B}^{-1}))$.
 6. Провести метрический анализ пространства параметров ученика и снизить его размерность.
-

Обучение байесовской модели ученика производится при помощи вариационного вывода на основе совместного правдоподобия данных:

$$\mathcal{L}(\mathcal{D}) = \log p(\mathcal{D}) = \log \int_{\mathbf{w} \in \mathbb{R}^k} p(\mathcal{D}|\mathbf{w}) \cdot p(\mathbf{w}) d\mathbf{w},$$

где $p(\mathbf{w})$ - априорное распределение параметров модели ученика, которое получается преобразованием из апостериорного распределения параметров учителя. С учетом нашего предположения, что оно является нормальным, обучение ученика сводится к решению задачи:

$$\hat{\mathbf{w}} = \arg \min_{\mu, \Sigma, \mathbf{w}} D_{KL}(p(\mathbf{w}|\mathcal{D}) || p(\mathbf{w})) - \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}).$$

Где второе слагаемое - логарифм правдоподобия выборки, а первое - KL-дивергенция между априорным и апостериорным распределением параметров ученика.

274 4.1 Вычисление ковариации

275 Вычисление ковариации будет производиться следующими способами, представленными
276 в статье [Chen et al., 2020].

Пусть $\mathbf{w}^* \in \mathbb{R}^d$ - истинный вектор параметров модели, то есть

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{W}} F(\mathbf{w})$$

$$F(\mathbf{w}) = \mathbb{E}_{\xi \sim \mathcal{D}} S(\mathbf{w}, \xi)$$

277 Где ξ - элемент, сэмплированный из обучающей выборки, S - функция ошибки.

При оптимизации параметров модели методом SGD с начальной точкой \mathbf{w}_0 происходит итеративный процесс:

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \nu_i \nabla S(\mathbf{w}_{i-1}, \xi_i)$$

278 Где ξ_i сэмплировано из обучающей выборки \mathcal{D} , $\nabla S(\mathbf{w}_{i-1}, \xi_i)$ - градиент функции ошибки
279 относительно весов модели.

280 В версии ASGS (averaged SGD), которая будет использоваться далее, в качестве ответа
281 возвращается $\overline{\mathbf{w}}_n = \frac{1}{n} \sum_{i=1}^n \mathbf{w}_i$.

282 Обозначим $A = \nabla^2 F(\mathbf{w}^*)$ - Гессиан матожидания ошибки, $S = \mathbb{E}[\nabla S(\mathbf{w}^*, \xi) \cdot \nabla S(\mathbf{w}^*, \xi)^\top]$
283 - ковариационная матрица $\nabla S(\mathbf{w}^*, \xi)$.

В статье упоминается, что при условиях на выпуклость F верно следующее асимптотическое равенство:

$$\sqrt{n}(\overline{\mathbf{w}}_n - \mathbf{w}^*) \rightarrow \mathcal{N}(0, A^{-1}SA^{-1})$$

284 Таким образом, состоятельная оценка матрицы ковариации $\sqrt{n}\overline{\mathbf{w}}_n$ это $A^{-1}SA^{-1}$.

Алгоритм 2 Оффлайн метод оценки ковариационной матрицы параметров

1. Обучить модель, получив приближение \mathbf{w}^* .
2. Приблизить матрицы A, S с помощью сэмплирования:

$$A_n = \frac{1}{n} \sum_{i=1}^n \nabla^2 S(\mathbf{w}^*, \xi_i), \quad S_n = \frac{1}{n} \sum_{i=1}^n \nabla S(\mathbf{w}^*, \xi_i) \cdot S(\mathbf{w}^*, \xi_i)^\top$$

3. Оценить ковариационную матрицу $cov(\sqrt{n}\bar{\mathbf{w}}_n) \approx A_n^{-1} S_n A_n^{-1}$.
-

Алгоритм 3 Онлайн метод оценки ковариационной матрицы параметров

В процессе обучения модели с помощью SGD:

1. Приблизить матрицы A, S с помощью n слагаемых:

$$A_n = \frac{1}{n} \sum_{i=1}^n \nabla^2 S(\mathbf{w}^*, \xi_i), \quad S_n = \frac{1}{n} \sum_{i=1}^n \nabla S(\mathbf{w}^*, \xi_i) \cdot S(\mathbf{w}^*, \xi_i)^\top$$

Примечание: здесь ξ_i - не сэмплируются, а берутся из алгоритма оптимизации во время SGD.

4.2 Вычислительный эксперимент

Решается задача восстановления временного ряда Accelerometer Motion Sense с помощью 2-х и 2-х слойных нейросетей.

Модели 2-х слойной нейросети архитектуры $100 - 3 - 100$ и 3-х слойной архитектуры $100 - 3 - 20 - 100$ в течение 50 эпох обучены на датасете Accelerometer Motion Sense, состоящем из 10000 замеров ускорения устройства. После обучения матрицы ковариаций нейронов модели вычисляются по алгоритму 2.

Отметим, что матрицы A_n, S_n вычислялись **отдельно** для каждого нейрона, так как мы работаем в предположении, что И теоретически это могло повлиять на

Эксперимент по восстановлению временного ряда Accelerometer Motion Sense произведен

5 *

Список литературы

Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693, 2022.

Cem Örneк and Elif Vural. Nonlinear supervised dimensionality reduction via smooth regular embeddings. *Pattern Recognition*, 87:55–66, 2019.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

RV Isachenko and VV Strijov. Quadratic programming feature selection for multicorrelated signal decoding with partial least squares. *Expert Systems with Applications*, 207:117967, 2022.

Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*, 30, 2017.

Андрей Валериевич Грабовой, Олег Юрьевич Бахтеев, and Вадим Викторович Стрижов. Определение релевантности параметров нейросети. *Информатика и её применения*, 13(2):62–70, 2019.

Андрей Валериевич Грабовой, Олег Юрьевич Бахтеев, and Вадим Викторович Стрижов. Введение отношения порядка на множестве параметров аппроксимирующих моделей. *Информатика и её применения*, 14(2):58–65, 2020.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

- 318 Nina Golyandina, Vladimir Nekrutkin, and Anatoly A Zhigljavsky. *Analysis of time series*
319 *structure: SSA and related techniques*. CRC press, 2001.
- 320 Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning:
321 Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, pages 89–95,
322 2021.
- 323 Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations*
324 *and Trends® in Machine Learning*, 12(4):307–392, 2019.
- 325 Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary
326 differential equations. *Advances in neural information processing systems*, 31, 2018.
- 327 Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting
328 sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by*
329 *Design in Distributed Systems*, pages 1–6, 2018.
- 330 Steven M Peterson, Zoe Steine-Hanson, Nathan Davis, Rajesh PN Rao, and Bingni W Brunton.
331 Generalized neural decoders for transfer learning across participants and recording modalities.
332 *Journal of Neural Engineering*, 18(2):026014, 2021.
- 333 Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P
334 Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based
335 brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.
- 336 Andrey Valerievich Grabovoy and Vadim V Strijov. Bayesian distillation of deep learning models.
337 *Automation and Remote Control*, 82:1846–1856, 2021.
- 338 Xi Chen, Jason D Lee, Xin T Tong, and Yichen Zhang. Statistical inference for model parameters
339 in stochastic gradient descent. 2020.