Метрический анализ пространства параметров глубоких нейросетей

Эрнест Р. Насыров, Вадим В. Стрижов

nasyrov.rr@phystech.edu

Исследуется проблема снижения размерности пространства параметров модели машинного обучения. Решается задача восстановления временного ряда. Для восстановления используются авторегресионные модели: линейные, автоенкодеры, реккурентные 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 матожидание и матрица ковариации с помощью методов *бутстрэпа* 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 (reccurent 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 эмбеддинга в низкоразмерное представление, а в [Cunningham and Yu, 2014] обсуждаюся 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 *векторы* параметров посредством скалярных произведений, то есть упускается из виду простая *структура* преобразования.

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

В данном исследовании мы будем рассматривать нейросети простейшей структуры - которые являются композицией линейных и простых нелинейных функций (функций активации). Их составной блок описан формой:

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

В методах OBS, OBD и в статье [Грабовой et al., 2019] элементы $\mathbf{W}_{ij}$ исследовались по-отдельности, как скаляры. Авторы работы предлагают изучать их как векторы-строки

$$\mathbf{w}_1, \ldots, \mathbf{w}_m : \mathbf{W} = \begin{pmatrix} \mathbf{w}_1^\mathsf{T} \\ \ldots \\ \mathbf{w}_m^\mathsf{T} \end{pmatrix}.$$

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

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

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

В работе исследуется положение случайных векторов параметров модели $\mathbf{w}_i$ в метрическом пространстве. С помощью методов бутстрэпа и вариационного вывода [Hastie et al., 2009] оцениваются их матожидания $\mathbf{e}_i = \mathsf{E}(\mathbf{w}_i)$ и ковариационные матрицы $\mathsf{D}(\mathbf{w}_i) = \mathbf{A}_i^{-1}$. 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. Точкам максимума куполов распределения соответствуют матожидания векторов, а их форма определяется матрицей ковариации $\mathbf{A}^{-1}$. Таким образом, чем ниже и шире «купол», тем больше дисперсия и наоборот.

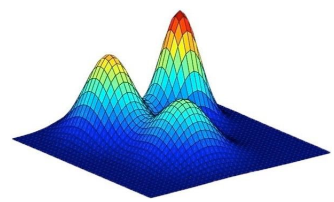


Рис. 1 Смесь гауссианов трех 2-х мерных векторов.

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.

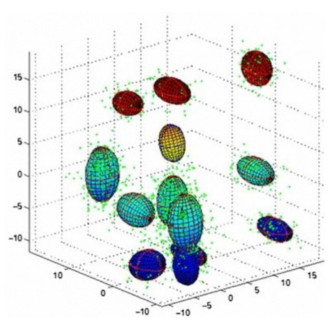


Рис. 2 Доверительные области 3-х мерных векторов.

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 (singular spectrum analysis)([Golyandina et al., 2001]), нелинейный PCA (principal component analysis), RNN (reccurent neural network)

([Bronstein et al., 2021]), VAE (variational autoencoder)([Kingma et al., 2019]) и Neural ODE (neural ordinary differential equations)([Chen et al., 2018]).

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

2 Problem statement

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

Definition 1. *Временное представление $\mathbf{x}_t = [\mathbf{s}_1^t, \ldots, \mathbf{s}_m^t]^\mathsf{T} \in \mathbb{R}^m$ 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 $\mathbf{X}_{t,h} = [\mathbf{x}_{t-h+1}, \ldots, \mathbf{x}_t]^\mathsf{T} \in \mathbb{R}^{h\times m}$.*

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

Definition 4. *Прогностическая модель $\mathbf{f}^{\mathsf{AR}} : \mathbb{R}^{h\times m} \to \mathbb{R}^{p\times m}$ 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 $\mathbf{f}^{\mathsf{AR}}$, 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 $\mathbb{S}$:

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

Длину горизонта планирования зафиксируем равной 1. Тогда прогностическая модель — это функция $f^{\mathsf{AR}} : \mathbb{S} \to \mathbb{R}$. 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 состоит в том, чтобы предсказать $\mathbf{s}^T$ с максимальной точностью. Предсказание происходит следующим образом:

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

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 $S = S(\mathbf{w}|\mathbf{s}, f)$:

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

В работе будет использоваться функция ошибки MSE, то есть

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

.

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

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

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

В качестве альтернативных моделей для восстановления временного ряда будем использовать 1) SSA, 2) двуслойную нейросеть с ортогональными линейными преобразованиями (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

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

Обозначим через

$$\lambda_1 \geqslant \ldots \geqslant \lambda_L \geqslant 0$$

собственные значения $\mathbf{S}$, а через

$$u_1, \ldots u_h$$

ортонормированную систему собственных векторов, соответствующую собственным значениям,

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

Тогда

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

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

$$\hat{\mathbf{X}} = \tilde{\mathbf{X}}_{I_1} + \cdots + \tilde{\mathbf{X}}_{I_m},\ I_1 \sqcup \cdots \sqcup I_m \subset \{1, \ldots, h\},\ \tilde{\mathbf{X}}_I = hank\big(\sum_{i\in I} \mathbf{X}_i\big).$$

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

2.2 Нейросеть

Модель двуслойной нейросети с ортогональными преобразованиями задается следующим образом:

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

где

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

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

2.3 RNN

Модель RNN задается следующим образом:

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

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

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

О входном временном ряде выдвинута гипотеза о том, что точки на фазовой траектории 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 $\hat{\mathbf{x}}_t$ — это матожидание точки фазовой траектории в момент времени t, а $\mathbf{B}$ — матрица ковариации.

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

Для метрического анализа пространства параметров выбираются набор множеств индексов $\mathcal{I}_1 \sqcup \dots \sqcup \mathcal{I}_k \subset \{1, \dots, len(\mathbf{w})\}$, рассматриваются соответствующие им подвектора параметров $\mathbf{w}_{\mathcal{I}_1}, \ldots, \mathbf{w}_{\mathcal{I}_k}$ Считается, что каждый $\mathbf{w}_{\mathcal{I}_j}$ соответствует «смысловой единице» модели. В нейросетевых моделях это строки матрицы линейного преобразования пространства параметров $\mathbf{W}$, которые обычно называют *нейронами*.

Для каждого $\mathbf{w}_{\mathcal{I}}$ оценивается матожидание $\hat{\mathbf{w}}_{\mathcal{I}}$ и ковариационная матрица $\mathbf{A}_{\mathcal{I}}$, 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.

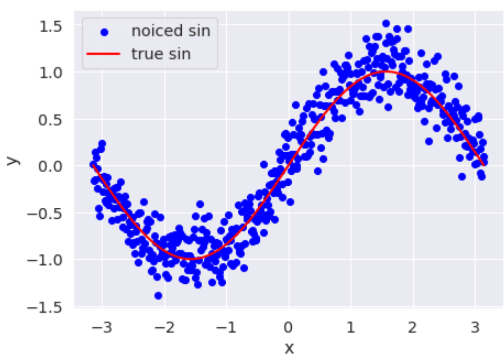
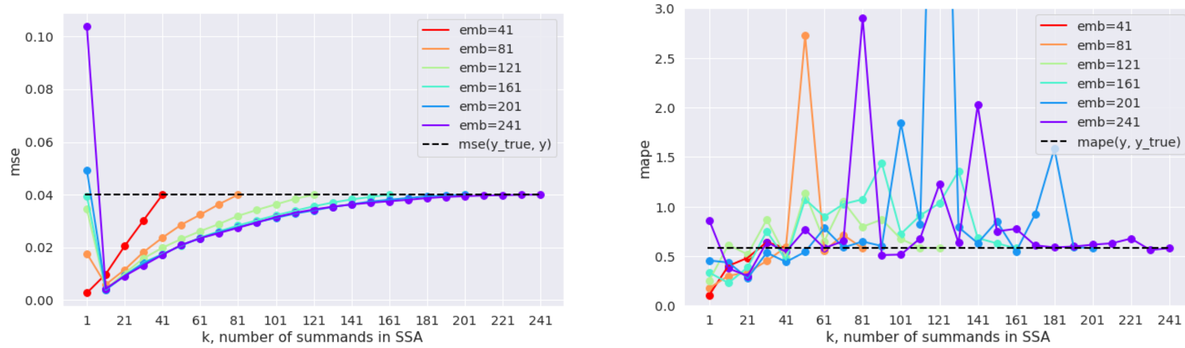


Рис. 3 Зашумленный синус.

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.4.

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 достатоно небольшом количестве слагаемых (< 20), что соответствует точке минимума всех графиков на левом рисунке. Чем больше компонет 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).



(*а*) MSE (*б*) MAPE

Рис. 4 Метрики для различных значений h и k. Черной линией выделена метрика между чистым и зашумленным синусом.

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. 5.

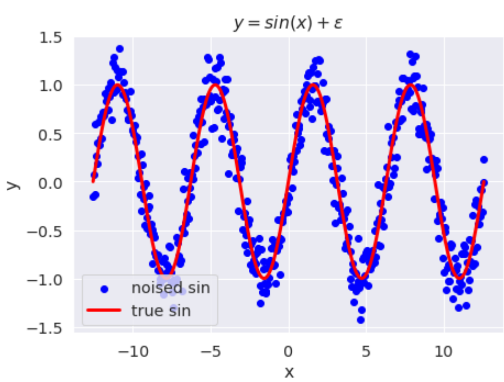


Рис. 5 Зашумленный синус.

С помощью нейросети решалась заадча 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 6.

Все параметры-нейроны на Рис. 6 имеют форму вытянутых эллипсов, расположенных на удалении друг от друга. Часть эллипсов вытянута в одну и ту же сторону (уменьшения оси OY). В целом все эти нейроны кажутся довольно независимыми и не имеющими выраженной структуры сообществ.

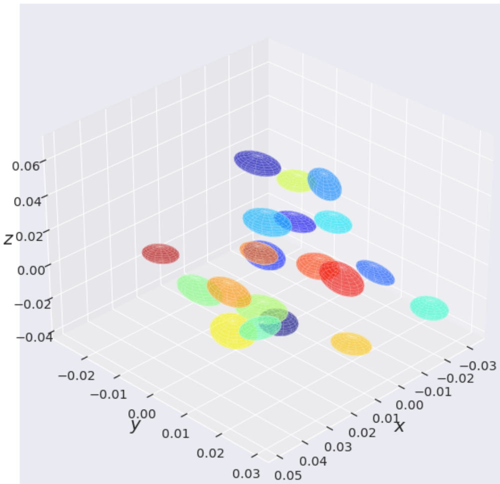


Рис. 6 Доверительные области 3-х мерных векторов двуслойной нейросети архитектуры 20-3-20.

В следующем эксперимента с нейросетью бралась архитектура $10 - 3 - 20$, которая означает, что на вход подаются $h_1 = 10$-мерные векторы, размерность скрытого пространства $d = 3$, а восстановить нужно $h_1 = 20$ моментов времени.

На случайных подвыборках обучающей выборки были обучены $N = 100$ нейросетей на 70 эпохах. Оценены матожидания и матрицы ковариаций столбцов параметра $\mathbf{W}_1 \in \mathbf{R}^{3\times 10}$. Полученные матрицы визуализированы в 3-х мерном пространстве Рис. 7.

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

Это частично подтверждает гипотезу о существовании структуры сообществ у нейронов.

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 внитри каждой частоты и затем pointwise convolution для смешение признаков).

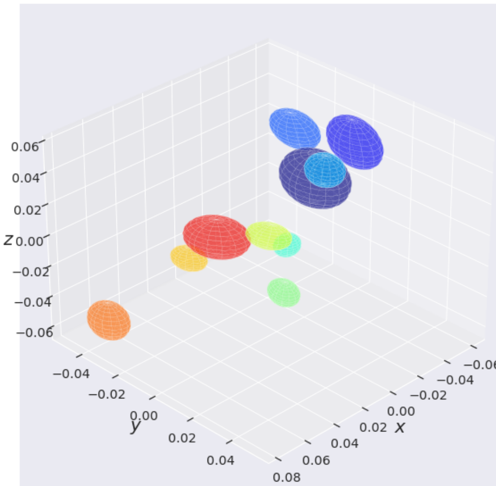


Рис. 7 Доверительные области 3-х мерных векторов двуслойной нейросети архитектуры 10-3-20.

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:

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

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

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

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

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

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

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

Мы считаем, что

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

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 $\overline{\mathbf{u}}, \mathbf{A}^{-1}$.

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 парметров ученика $p(\mathbf{w}|\mathcal{D}) = p(\mathbf{N}(\overline{\mathbf{w}}_t, \mathbf{B}^{-1}))$.

6. Провести метричекий анализ пространства параметров ученика и снизить его размерность.

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

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

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:

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

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

4.1 Вычисление ковариации

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

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

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

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

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

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

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

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

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

Обозначим $A = \nabla^2 F(\mathbf{w}^*)$ - Гессиан матожидания ошибки, $S = \mathsf{E}\left[\nabla \mathsf{S}(\mathbf{w}^*, \xi) \cdot \nabla \mathsf{S}(\mathbf{w}^*, \xi)^\mathsf{T}\right]$ - ковариационная матрица $\nabla S(\mathbf{w}^*, \xi)$.

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

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

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

Алгоритм 2 Оффлайн метод оценки ковариационной матрицы параметров

1. Обучить модель, получив приближение $\mathbf{w}^*$.

2. Приблизить матрицы A, S с помощью сэмплирования:

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

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

Алгоритм 3 Онлайн метод оценки ковариационной матрицы параметров

В процессе обучения модели с помощью SGD:

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

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

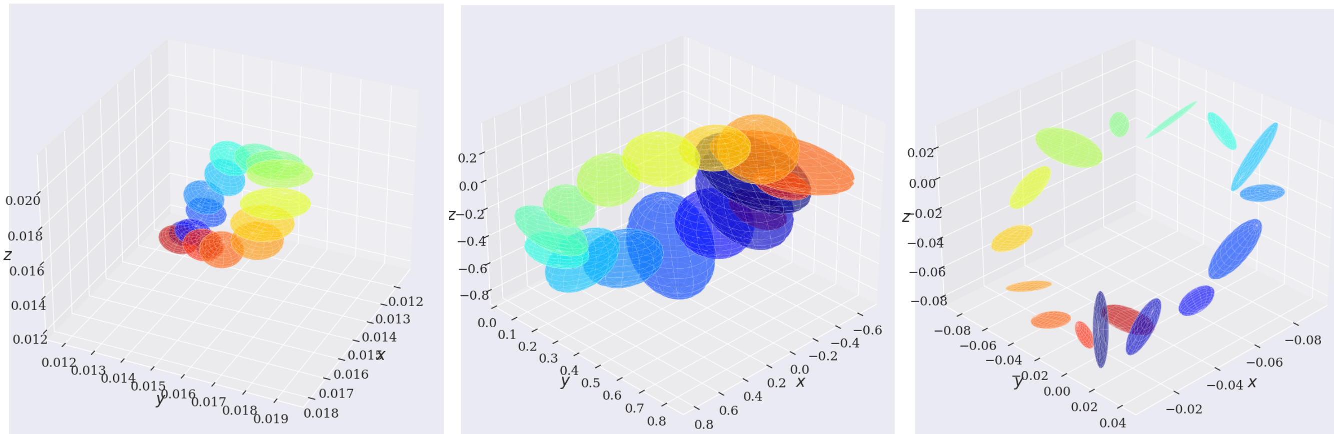
Примечание: здесь ξ_i - не сэмплируются, а берутся из алгоритма оптимизации во время SGD.

4.2 Вычислительный эксперимент

4.3 Оценка ковариаций

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

Модели 2-х слойной нейросети архитектуры $100 - 3 - 100$ в течение 50 эпох обучены на датасете Accelerometer Motion Sense, состоящем из 10000 замеров ускорения устройства. После обучения матрицы ковариаций нейронов модели вычислялись с помощью техники Bootstrap, по алгоритму 2 (техника Hessian), с помощью обучения байесовской нейросети (техника Bayes). Результаты представлены на Рис.8.



(а) Bootstrap (б) Hessian (в) Bayes

Рис. 8 Первые 16 нейронов нейросетей архитектуры $100 - 3 - 100$, обученных на задаче восстановления временного ряда Motion Sense.

Структура нейронов одинакова у всех 3 методов: они все расположены «по кругу». С учетом того, что период временного ряда - 100 измерений, то очевидно, что размер выходного слоя нейросети можно уменьшить до 16. Значительно различается масштаб: оценки методом Bootstrap имеют наименьшие дисперсии и разброс нейронов, а по методу Hessian - наибольшие. Также видно, что метода Hessian и Bayes дает плоские нейроны (одно из собственных значений ковариационной матрицы во много раз меньше 2 других), в отличие от метода Bootstrap. Это дает потенциальную возможность произвести линейные преобразования нейронов скрытого слоя, чтобы сократить размер скрытого пространства с 3 до 2 и представляет интерес для дальнейших исследований.

4.4 Снижение размерности с помощью кластеризации.

На 3 датасетах: MNIST, Iris flower и Motion Sence решались задачи классификации и регрессии с помощью полносвязных нейронных сетей. Архитектуры: $784 - 8 - 100 - 10$, $4 - 128 - 3$ и $100 - 10 - 100$ соответственно.

Оценки ковариаций и матожиданий нейронов проведены с помощью техники bootstrap: были натренированы 100 моделей с помощью SGD и подсчитаны несмещенные оценки ковариаций и дисперсий нейронов по формулам:

$$\mathsf{E}[\mathbf{w}] = \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^t, \quad cov(\mathbf{w}_i, \mathbf{w}_j) = \frac{\sum\limits_{t=1}^{n}(\mathbf{w}_i^t - \overline{\mathbf{w}_i})(\mathbf{w}_j^t - \overline{\mathbf{w}_j})}{n-1}$$

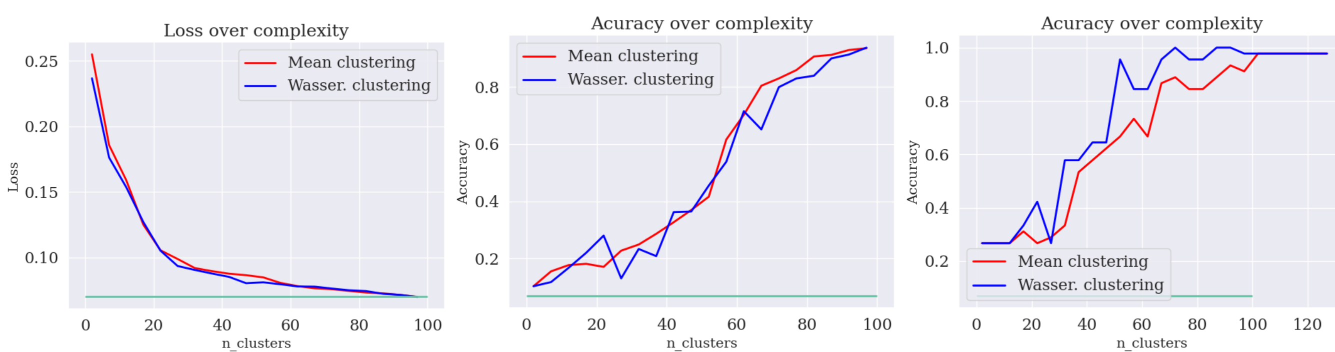
Кластеризация нейронов проводилась с помощью алгоритма $k - means$ двумя способами: в первом расстояние между нейронами считалось, как евклидово расстояние между их средними, а во втором как расстояние Вассерштайна между двумя нормальными распределениями:

$$\rho_1(\mathbf{w_1}, \mathbf{w_2}) = ||\mathbf{m}_1 - \mathbf{m}_2||_2^2$$

$$\rho_2(\mathbf{w_1}, \mathbf{w_2}) = ||\mathbf{m}_1 - \mathbf{m}_2||_2^2 + Tr(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 - 2(\mathbf{\Sigma}_1^{0.5}\mathbf{\Sigma}_2\mathbf{\Sigma}_1^{0.5})^{0.5})$$

После кластеризации каждый нейрон заменялся на центроид своего кластера и производились замеры качества: MSE-ошибка для задачи регрессии и Accuracy для классификации.

Результаты экспериментов представлены на Рис.9.



(*а*) MotionSense. 10-и мерные нейроны. (*б*) MNIST. 8-ми мерные нейроны. (*в*) IRIS. 4-х мерные нейроны.

Рис. 9 Зависимость точности модели от сложности на 3 датасетах. Синяя кривая соответствует кластеризации по мерике Вассерштайна, а красная — по евклидовой метрике.

На графиках показана зависимость точности модели от ее сложности. Чем большее число кластеров выделено, тем больше параметров имеет модель и тем она сложнее и, соответственно, растет качество предсказаний. Видим, что оба способа кластеризации показывают одинаковые результаты на первых двух датасетах, а на датасете IRIS кластеризация Вассерштайна показала лучше. Скорее всего это связано с размерностью нейронов: чем она меньше, тем более точно можно оценить ковариационную матрицу и тем лучше получающиеся кластеры.

5 Заключение

В работе исследовано, как выглядит пространство нейронов нейросетей при задаче восстановления временного ряда. На временном ряде MotionSence показано, что 3-х мерные нейроны имеют регулярную структуру и располагаются «по кругу» с периодом 16, что в 5 раз меньше периода самого временного ряда, что представляет возможность уменьшить размер скрытого пространства сети более чем в 5 раз.

Также в работе исследована возможность снижения размерности пространства параметров нейросетей с помощью кластеризации нейронов на 3 датасетах разной природы: MotionSence - скалярный временной ряд, решалась задача регрессии; IRIS - табличные данные, решалась задача классификации; MNIST - картинки, решалась задача классификации.

Показано, что на датасете IRIS можно удалить до 50% нейронов, обеспечивая в то же время более чем 90% точность классификации. На остальных датасетах эффект не такой значительный.

Будущие исследования будут посвящены изучению сжатия больших, промышленных нейросетей и особое внимание будет уделено сжатию сетей, восстанавливающих временные ряды, как потенциально имеющих избыточное количество параметров.

Также будут разработаны новые методы оценки матриц ковариации параметров: с помощью низкоранговых байесовских сетей, полубайесовских и блочно-диагонально-байесовских.

6 *

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 Örnek 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.

Nina Golyandina, Vladimir Nekrutkin, and Anatoly A Zhigljavsky. *Analysis of time series structure: SSA and related techniques*. CRC press, 2001.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, pages 89–95, 2021.

Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*, pages 1–6, 2018.

Steven M Peterson, Zoe Steine-Hanson, Nathan Davis, Rajesh PN Rao, and Bingni W Brunton. Generalized neural decoders for transfer learning across participants and recording modalities. *Journal of Neural Engineering*, 18(2):026014, 2021.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Andrey Valerievich Grabovoy and Vadim V Strijov. Bayesian distillation of deep learning models. *Automation and Remote Control*, 82:1846–1856, 2021.

Xi Chen, Jason D Lee, Xin T Tong, and Yichen Zhang. Statistical inference for model parameters in stochastic gradient descent. 2020.