

# Метрический анализ пространства параметров глубоких нейросетей.

Насыров Р.Р.

МФТИ

19 мая 2023 г.

# Введение

- Высокоразмерные данные - видео, звук - избыточны.
- Модели тяжело обучаются на избыточных данных, часто переобучаются.
- Нужно бороться с избыточностью и переобучением.

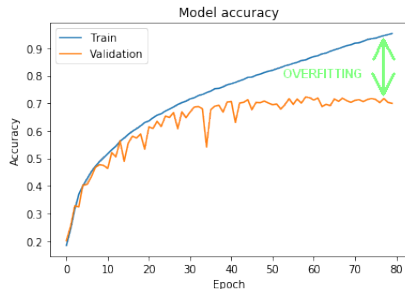


Рис.: Переобучение.

# Методы улучшения обучения

- Методы снижения размерности входных данных:
  - ▶ PCA
  - ▶ Quadratic Programming Feature selection
  - ▶ Neural Autoencoders
- Выбор оптимальной структуры модели:
  - ▶ Optimal Brain Surgeon
  - ▶ Correlational analysis
  - ▶ Weights freezing

# Цель исследования

**Цель исследования:** создать метод снижения размерности пространства параметров глубоких нейросетей.

**Задача:** построить алгоритм изменения нейронов, который

- Оценивает дисперсию нейронов
- Определяет релевантные нейроны, фильтрует выбросы
- Выделяет сообщества нейронов на основе дисперсий
- Находит центры сообществ и заменяет веса нейронов сообщества на вес центра сообщества

# Отбор нейронов сети метрическими методами.

Модель двуслойной нейросети:

$$f(x) = \sigma(W_1^T \cdot \sigma(W_2^T x + b_1) + b_2)$$

Модель RNN:

$$h_t = \sigma(W \cdot h_{t-1} + V \cdot x_t),$$

$$s_{t+1} = \tanh(w_o^T \cdot h_t)$$

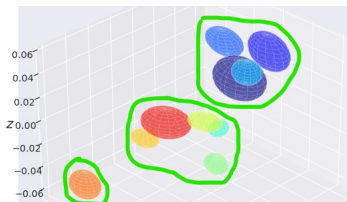


Рис.: Доверительные области нейронов сети 10-3-20.



Рис.: Схема экспериментов.

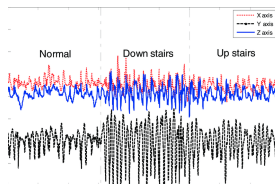


Рис.: Показания акселерометра ходьбы.

# Литература

- Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex Intelligent Systems*, 8(3):2663–2693, 2022.
- Cem Ornek and Elif Vural. Nonlinear supervised dimensionality reduction via smooth regular embeddings. *Pattern Recognition*, 87:55–66, 2019.
- RV Isachenko and VV Strijov. Quadratic programming feature selection for multicorrelated282 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.
- 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.

# Постановка задачи

- Решается задача авторегрессионного декодирования.
- Обозначим множество всех одномерных временных рядов через  $\mathbb{S}$ :

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

- Авторегрессионная модель  $f^{\text{AR}} : \mathbb{S} \rightarrow \mathbb{S}$  восстанавливает значения временного ряда.
- Рассматриваются feed-forward нейросети:

$$f(x) = \sigma(W_1^T \cdot \sigma(W_2^T \cdot \dots (W_n^T \cdot x + b_n) \cdots + b_2) + b_1)$$

- Гипотеза о порождении данных:  $x_t = [s_t, \dots, s_{t+h-1}] \sim \mathcal{N}(\hat{x}_t, B)$ , где  $\hat{x}_t$  — это матожидание точки фазовой траектории в момент времени  $t$ , а  $B$  — матрица ковариации.
- Гипотеза о распределении параметров модели:

$$W = \begin{pmatrix} w_1^T \\ \dots \\ w_n^T \end{pmatrix} : w_i \sim \mathcal{N}(\mu_i, A_i)$$

# Постановка задачи

- Для каждой матрицы  $W_i$  требуется построить матрицу  $\widetilde{W}_i$  меньших размеров, сохраняя точность восстановления ряда на приемлемом уровне.
- Модель  $f = f(w, s)$ ,  $w \in \mathbb{W}$  выбирается из некоего параметрического семейства нейросетей. Параметры модели выбираются таким образом, чтобы минимизировать функцию ошибки  $S = S(w|s, f)$ :

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

В работе будет использоваться функция ошибки MSE, то есть

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

## Решение задачи, теоретическая часть

- Задан временной ряд  $s = (s_1, \dots, s_n) \in \mathbb{R}^n$ .
- Задана модель учителя в виде суперпозиций линейных и нелинейных преобразований:

$$f = \sigma \circ U_T \circ \sigma \circ U_{T-1} \circ \dots \circ U_2 \circ \sigma \circ U_1,$$

где  $T$  — число слоев модели учителя (от 5 до 8 у применяемых далее моделей),  $\sigma$  — функция активации,  $U_t$  — матрица линейного преобразования. Матрицы  $U_t$ , параметры модели учителя, соединяются в вектор параметров  $u$  модели учителя  $f$ :

$$u = \text{vec}([U_T, U_{T-1}, \dots, U_1]).$$

## Решение задачи, теоретическая часть

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

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

- В качестве модели ученика используется байесовская нейронная сеть, задаваемая аналогично модели учителя:

$$g = \sigma \circ W_L \circ \sigma \circ W_{L-1} \circ \dots \circ W_2 \circ \sigma \circ W_1.$$

Мы считаем, что

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

---

## Algorithm 1 Алгоритм снижения размерности

---

- 1 Зафиксировать структуру модели ученика  $g$  и учителя  $f$ , то есть параметры  $T, L$  и размеры матриц  $U_t, W_t$ .
  - 2 Обучить модель учителя  $f$ .
  - 3 Оценить апостериорное распределение параметров учителя  $p_{ps}(u)$ , то есть параметры  $\bar{u}, A^{-1}$ .
  - 4 Назначить параметрам ученика априорное распределение  $p(w) \sim p_{ps}(u)$ .
  - 5 Обучить байесовскую сеть ученика на ответах учителя, получив апостериорное распределение параметров ученика  $p(w|\mathcal{D}) = p(N(\bar{w}_t, B^{-1}))$ .
  - 6 Провести метрический анализ пространства параметров ученика и снизить его размерность.
-

## Решение задачи, теоретическая часть

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

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

где  $p(w)$  - априорное распределение параметров модели ученика, которое получается преобразованием из апостериорного распределения параметров учителя.

- Обучение ученика сводится к решению задачи:

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

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

## Решение задачи, теоретическая часть

Выделение структуры сообществ проводится с помощью алгоритма «k-means clustering»:

- Дан набор наблюдений  $(x_1, \dots, x_n), x_i \in \mathbb{R}^d$ .
- Алгоритм «k-means» делит наблюдения на  $k$  классов  $\mathcal{S} = (S_1, \dots, S_k)$ , минимизируя внутриклассовую дисперсию:

$$\sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \rightarrow \min_{\mathcal{S}}.$$

- Где внутриклассовое среднее

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$$

# Вычислительный эксперимент

**Цель эксперимента:** визуализировать пространство нейронов сети в задаче восстановления временного ряда.

**Данные:**

- Показания акселерометра, записанные с частотой 50Гц во время 6 видов активностей у 20 людей в течение 2 минут.

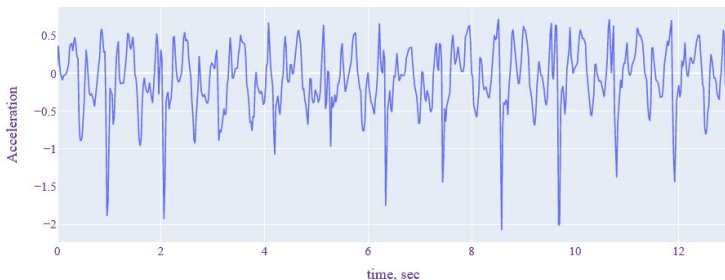


Рис.: Данные акселерометра.

# Способы вычисления ковариации параметров нейрона

- Bootstrap: обучаются  $N$  одинаковых моделей на случайной подвыборке тренировочных данных. Ковариация нейрона  $w \in \mathbb{R}^n$  считается по формуле:

$$\text{cov}(w_i, w_j) = \frac{1}{N-1} \sum_{t=1}^N (w_i^t - \bar{w}_i)(w_j^t - \bar{w}_j)$$

- Bayes NN: обучается байесовская нейросеть, в которой нейроны  $w \in \mathbb{R}^n$  параметризуются с помощью матрицы  $A \in \mathbb{R}^{n \times k}$ ,  $\text{rk}(A) = k$ ,  $AA^T = \text{cov}(w)$  и вектора-матожидания  $\bar{w}$ .
- Hessian: по обученной модели оцениваем гессиан и градиент ошибки по параметрам модели. По ним вычисляем матрицу:

$$H_n = \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla^2 S(w^*, \xi)], \quad G_n = \mathbb{E}_{\xi \sim \mathcal{D}}[\nabla S(w^*, \xi) \cdot \nabla S(w^*, \xi)^T]$$

Оценка:  $\text{cov}(\sqrt{n}\bar{w}_n) \approx H_n^{-1} G_n H_n^{-1}$ .

# Визуализация пространства нейронов 3 методами

Визуализация доверительных интервалов первых 16 нейронов нейросети архитектуры 100 – 3 – 100, обученной на датасете MotionSense.

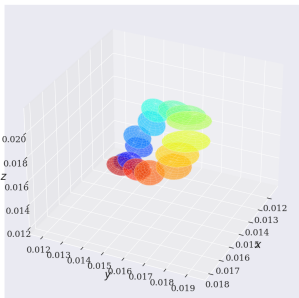


Рис.: Bootstrap

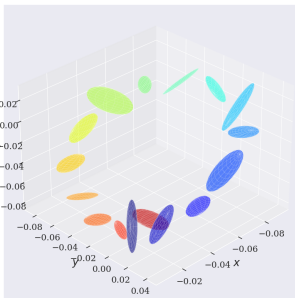


Рис.: Bayes NN

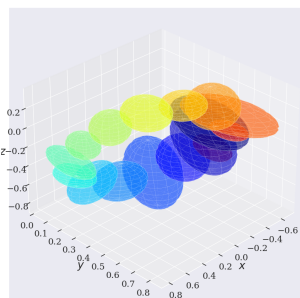


Рис.: Hessian

- нейроны расположены по кругу.
- возможно снижение размерности пространства с 100 до 16.

## Снижение размерности пространства параметров с помощью кластеризации методом «k-means».

- На 3 датасетах: MNIST, Iris flower и Motion Sence решались задачи классификации и регрессии с помощью полносвязных нейронных сетей.
- Параметры распределения оцениваются с помощью bootstrap.
- 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:

$$\rho_1(w_1, w_2) = \|m_1 - m_2\|_2^2$$

$$\rho_2(w_1, w_2) = \|m_1 - m_2\|_2^2 + \text{Tr}(A_1 + A_2 - 2(A_1^{0.5} A_2 A_1^{0.5})^{0.5})$$

# Точности модели VS количество параметров.

На графиках показана зависимость точности модели от ее сложности. Красная кривая - кластеризация по 1 метрике, синяя - по 2-й. В скобках подписана размерность нейронов.

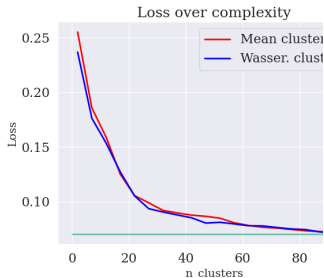


Рис.: MotionSense (10).

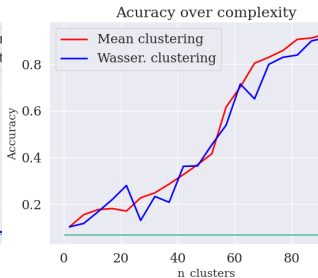


Рис.: MNIST (8).

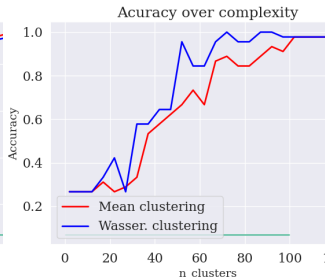


Рис.: IRIS (4).

- Учет ковариации улучшает качество модели при небольшой размерности нейронов.
- На датасете IRIS удастся сократить в 2 раза количество параметров, сохраняя точность более 90%.

# Выводы

- При восстановлении временных рядов нейроны имеют регулярную структуру, что допускает значительное сжатие количества параметров.
- Разные методы оценки параметров распределения дают разные численные результаты, но схожие по структуре (по «кругу»).
- Кластеризация по метрике Вассерштайна работает не хуже, а иногда даже значительно лучше, чем по обычной метрике.
- Это подтверждает актуальность задачи оценки ковариации нейронов и рассмотрения нейрона как одного целого.

## Следующие шаги

- Сжатие больших нейросетей.
- Разработка методов эффективной оценки матрицы ковариации параметров с помощью низкоранговых байесовских сетей, полубайесовских и блочно-диагонально-байесовских.
- Сравнение с известными методами снижения пространства параметров: OBS, OBD и других.