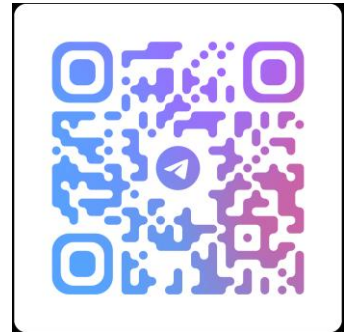


Основы практического использования нейронных сетей.

Лекция 1. Основные этапы разработки НС.

Дмитрий Буряк.
к.ф.-м.н.
dyb04@yandex.ru



Основные темы спецкурса

- ❑ Построение и обучение НС
 - Оптимизация гиперпараметров НС
 - Выбор функции ошибки
 - Анализ, оптимизация обученной сети
- ❑ Особенности построения глубоких НС
 - Алгоритмы обучения
 - Проблемы при обучении
 - Методы регуляризации
- ❑ Обзор современных архитектур глубоких нейронных сетей.
- ❑ Программные средства проектирования и реализации НС.
- ❑ Практическое задание

Ссылки на спецкурс:

- http://sqi.cs.msu.ru/learning/spec_courses/Neural_Nets_Advanced_2025
- Телеграм-канал



Литература

- ❑ <http://www.deeplearningbook.org/>
- ❑ <http://cs231n.github.io/>
- ❑ <http://neuralnetworksanddeeplearning.com/index.html>
- ❑ Франсуа Шолле. Глубокое обучение на Python, 2018.
Francois Chollet. Deep Learning with Python, 2nd edition, 2021
- ❑ Eli Stevens, et al., Deep Learning with PyTorch, 2020.

Практические принципы разработки НС

- ❑ Для успешного обучения недостаточно знания библиотек
 - Необходимо понимать, как работает каждый применяемый инструмент
- ❑ Сложно диагностировать ошибки при обучении НС
 - Многие ошибки не приводят к отсутствию сходимости при обучении
 - Похожее поведение наблюдается при неоптимальных значениях параметров

Основные этапы разработки ИС

- ❑ Определение задачи
- ❑ Выбор целевых метрик
- ❑ Сбор и анализ данных
- ❑ Разработка скелета конвейера предобработка-обучение-тестирование
 - Использование простейших архитектур
 - Цель: реализация и отладка инфраструктуры для обучения + получение начальных оценок целевых метрик
- ❑ Обучение
 - Анализ результатов обучения
 - Стратегия «переобучение-регуляризация»
- ❑ Тонкая настройка: оптимизация гиперпараметров
- ❑ Финальное [длительное] обучение

Определение задачи

- ❑ Входные данные: 1D вектор, 2D данные, последовательность,...
- ❑ Тип задачи
 - Бинарная классификация: спам-детектор
 - Многоклассовая классификация: распознавание команд
 - Классификация со множеством меток для каждого примера: поиск изображений
 - Регрессия: определение жизненных показателей человека (пульс, давление)
 - ...
- ❑ Выходные данные.

Выбор целевых метрик

❑ Показатели, которые определяют эффективность полученного решения.

Могут быть недифференцируемыми по весам НС.

❑ Классификация

- Точность
- F1-мера
- Площадь под ROC

		POSITIVE	NEGATIVE
ACTUAL VALUES	POSITIVE	TP	FN
	NEGATIVE	FP	TN

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

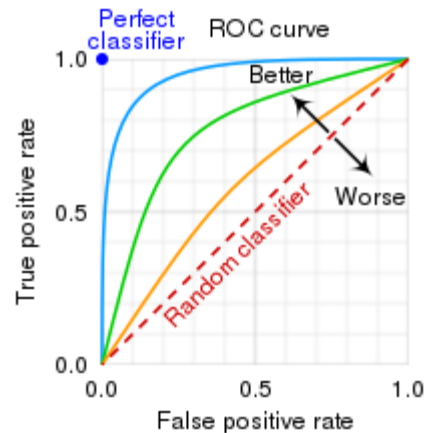
$$F1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

❑ Регрессия

- Близость по мере

❑ Выбор сценария оценки эффективности

- Использования валидационной выборки
- Кроссвалидация



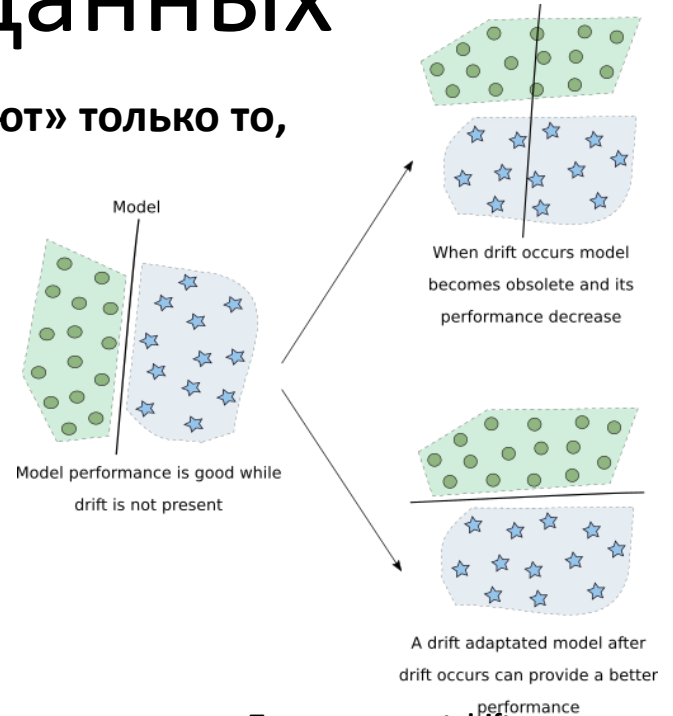
Сбор и анализ данных

❑ Алгоритмы машинного обучения «запоминают» только то, что было в обучающих данных.

❑ Основные свойства данных:

- Объем
- Репрезентативность: соответствие обучающих и продуктовых данных, data drift, concept drift, ...
- Качество разметки

❑ Анализ данных: статистика меток, распределение элементов данных, поиск закономерностей, корреляций, поиск дубликатов, выбросов...

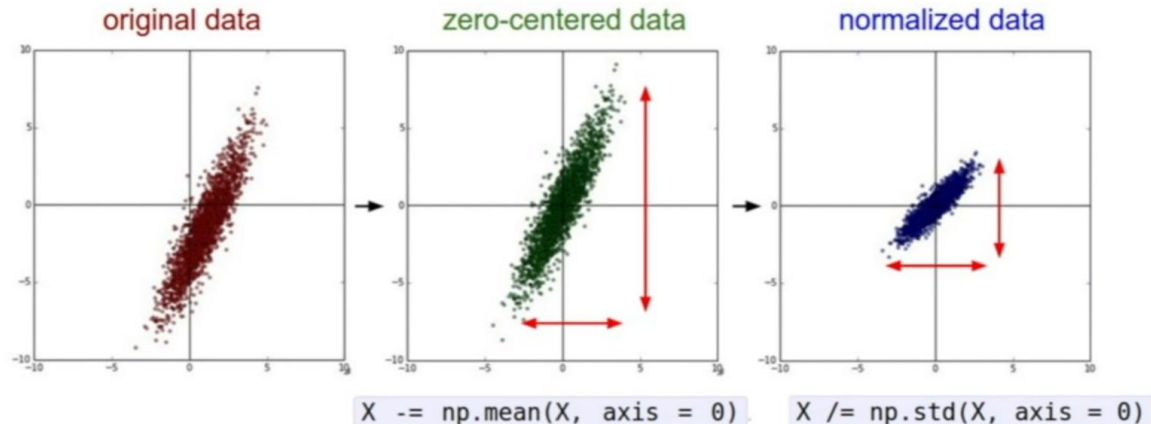


Пример concept drift

(<https://www.kdnuggets.com/2019/12/ravages-concept-drift-stream-learning-applications.html>)

Предобработка данных

- ❑ Векторизация данных
- ❑ Нормализация данных
 - Значения из небольшого интервала: $[0,1]$, $[-1,1]$, ...
 - Одинаковые диапазоны для каждого признака.
- ❑ Обработка пропущенных значений, выбросов



Низкая эффективность обучения НС

- Низкая эффективность = большая ошибка на тестовых данных.
- Проблемы с данными;
- Несоответствие архитектуры НС сложности задачи;
- Неоптимальные значения гиперпараметров;
- Переобучение;
- Ошибки в реализации.

Обозначения

- ❑ S_{train} – обучающая выборка; S_{test} – тестовая выборка;
- ❑ E_{train} – ошибка на обучающей выборке, E_{test} – ошибка на тестовой выборке, E_{goal} – целевое значение ошибки.

- ❑ $E_{test} > E_{goal}$

Анализ ошибки на обучающей выборке

$E_{train} > E_{goal}$

- Увеличить размер НС;
- Алгоритм обучения
- Оптимизировать значения гиперпараметров алгоритма обучения
- «Затухание» градиента: batch normalization, residual connection,...
- Анализ качества исходных данных
 - низкое значение сигнал-шум;
 - ошибки алгоритма предобработки;
 - недостоверные референсные значения;
 - несбалансированная выборка.

Анализ ошибки на тестовой выборке

$E_{train} < E_{goal}$ и $E_{test} > E_{goal}$

Увеличить размер S_{train} ;

Уменьшить размер НС;

Регуляризация, dropout, аугментация, ...;

Подбор алгоритма обучения;

Несоответствие S_{train} и S_{test} .

Вопросы

- ❑ Основные свойства данных при их использовании для обучения и тестирования НС?
- ❑ В чем отличие функции потерь от целевой метрики, используемой для оценки эффективности НС?
- ❑ Что может быть причиной, когда ошибка на обучающей выборке превосходит целевое значение ошибки?
- ❑ Какие действия следует предпринимать, чтобы добиться уменьшения ошибки на тестовой выборке?