

УПРАВЛЕНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

вводная лекция, экосистема Python, математика

А.В. Макаренко

`avm@rdcn.ru`

Научно-исследовательская группа «Конструктивная Кибернетика»
Москва, Россия, www.rdcn.ru

Институт проблем управления РАН
Москва, Россия

Учебный курс – Лекция 1

06 февраля 2020 г.

ИПУ РАН, Москва, Россия

- ➊ Предисловие к курсу
- ➋ Об авторе
- ➌ Общие положения
- ➍ Основные дисциплины курса
- ➎ Python
- ➏ Библиотеки и инструменты Python
- ➐ Открытые датасеты
- ➑ Векторно-матричные преобразования
- ➒ Заключение

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

О чём говорит название?

О чём говорит название?

УПРАВЛЕНИЕ – целенаправленный перевод (переход) системы Σ посредством управляющего воздействия U из одного состояния S_b в другое – требуемое S_e (определённое целями управления W_u).

О чём говорит название?

УПРАВЛЕНИЕ – целенаправленный перевод (переход) системы Σ посредством управляющего воздействия U из одного состояния S_b в другое – требуемое S_e (определённое целями управления W_u).

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

О чём говорит название?

УПРАВЛЕНИЕ – целенаправленный перевод (переход) системы Σ посредством управляющего воздействия U из одного состояния S_b в другое – требуемое S_e (определённое целями управления W_u).

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

Эксперт vs эрудит

Эксперт – это тот, кто знает всё о чём-то малом, эрудит – это тот кто знает мало, зато обо всём.. *Ричард Хэмминг.*



Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Общая информация

- Основатель и руководитель научно исследовательской группы «Конструктивная Кибернетика»
- Заведующий лабораторией 77 «Вычислительная Кибернетика» ИПУ РАН
- Директор Центра технологий искусственного интеллекта ИПУ РАН
- Старший научный сотрудник ИПУ РАН
- Кандидат технических наук по специальности 05.13.01 «Системный анализ, управление и обработка данных»
- IEEE Signal Processing Society Membership
- IEEE Computational Intelligence Society Membership

avm@rdcn.ru

www.rdcn.ru

Научные интересы

- Анализ структуры сложных динамических процессов, предсказуемость
- Обнаружение, классификация и диагностика не вполне наблюдаемых объектов (паттернов)
- Синхронизация и самоорганизация в нелинейных и хаотических системах
- Моделирование экономических, финансовых, социальных и биофизических систем и процессов
- Исследования в области конвергенции Data Science, Nonlinear Dynamic и Network-Centric

Инженерные навыки

- Machine Learning, Deep Learning, Data Mining, Signal Processing, High Performance Computing
- Языки программирования: C/C++, Julia, Python, R, Wolfram Language
- Технологии: MPI, OpenMP, Cilk Plus, CUDA
- Фреймворки и библиотеки: Keras, FaceBook PyTorch, Google TensorFlow, MS CNTK, Berkeley Caffe, MXNet, Intel MKL, Intel MPP, ...

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Программа курса I

#	Темы лекционных занятий	Содержание
1	Цели и задачи курса, основные разделы	Вводное занятие, определение механики курса, его наполнения и акцентов.
2	Язык программирования Python, основные библиотеки и инструменты	Основы языка Python 3.6. Дистрибутив Anaconda. Notebook Jupyter. IDE Spyder. Библиотеки Numpy, Matplotlib, Scipy, Pandas, Scikit-learn. Базовые дата-сеты.
3	Линейная алгебра, теория вероятностей и мат. статистика, случайные процессы, численные методы оптимизации	Базовые векторно-матричные операции: сложение и умножение. Собственные числа и вектора матриц, нормы. SVD разложение. Deskриптивная статистика, стат. гипотезы и критерии. Корреляционный и спектральный анализ. Разведочный анализ данных. Элементы языка R. IDE RStudio.
4	Искусственный интеллект, интеллектуальные задачи и системы	Определение, и проблематика ИИ, основные подходы к решению задач. Классификация интеллектуальных задач. Типы и виды интеллектуальных систем.

Программа курса II

#	Темы лекционных занятий	Содержание
5	Нелинейная динамика, хаос, дискретные отображения	Нелинейные динамические системы, их классификация. Аттракторы. Хаос и его критерии. Синхронизация. Анализ дискретных динамических систем в форме отображений. Прикладные аспекты задач анализа хаоса.
6	Кибернетика, задачи управления, наблюдаемость, идентифицируемость, управляемость, адаптивность.	Кибернетика в широком смысле. Классификация задач и систем управления. Понятия внешней среды и помех. Основные этапы анализа и синтеза систем управления.
7	Машинное обучение, Deep Learning, глубокие нейронные сети, фреймворк Keras	Основные задачи и методы машинного обучения. Режимы обучения. Функции потерь. Измерение качества моделей. Основная парадигма и мейнстрим глубокого обучения. Основные типы глубоких нейронных сетей: MLP, CNN, RNN (LSTM). Комбинация типов нейросетей и мультимодальные задачи. Возможности и ограничения глубоких нейронных сетей, проблематика их синтеза, обучения и встраивания в реальные бизнес-процессы. Работа с фреймворком Keras.

Программа курса III

#	Темы лекционных занятий	Содержание
8	Signal Processing, задачи и классические методы обработки временных рядов	Основные подходы и методы обработки сигналов: свёртка, фильтрация, Фурье-анализ, вейвлет-анализ. Статистические методы: обнаружение, различение, оценивание, прогнозирование сигналов.
9	Решение задач обработки временных рядов методами Deep Learning	Распознавание, оценивание, предсказание многомерных динамических процессов посредством глубоких свёрточных и рекуррентных (LSTM) нейронных сетей. Особенности формирования наборов исходных признаков, структуры входных данных и архитектуры сетей. Анализ предобученных моделей.

Программа курса III

#	Темы лекционных занятий	Содержание
8	Signal Processing, задачи и классические методы обработки временных рядов	Основные подходы и методы обработки сигналов: свёртка, фильтрация, Фурье-анализ, вейвлет-анализ. Статистические методы: обнаружение, различение, оценивание, прогнозирование сигналов.
9	Решение задач обработки временных рядов методами Deep Learning	Распознавание, оценивание, предсказание многомерных динамических процессов посредством глубоких свёрточных и рекуррентных (LSTM) нейронных сетей. Особенности формирования наборов исходных признаков, структуры входных данных и архитектуры сетей. Анализ предобученных моделей.

Обучение vs Тренировка

Обучение направлено на то, что, когда и зачем делать, а тренировка – как это делать. *Ричард Хэмминг.*

Программа курса IV

#	Темы «handmade» занятий	Содержание
1	Разведочный анализ данных	Дескриптивная статистика, заполнение пропусков, структура данных, статистические гипотезы, наблюдаемость систем
2	Анализ дискретных динамических систем	Временная и частотная области, динамика отображений, бифуркации, синхронизация, самоорганизация, идентифицируемость систем
3	Анализ структуры сигналов	Временная и частотная области, корреляционные характеристики, прогнозирование процессов
4	Задачи и алгоритмы машинного обучения	Классы задач, информативные признаки, основные shallow алгоритмы (логистическая регрессия, Random Forest, SVM, ...), стратегии обучения, функции потерь, регуляризация
5	Глубокие нейронные сети	Свёрточные сети, рекуррентные (LSTM) сети, структура входных данных, архитектуры сетей, стратегии обучения, функции потерь, регуляризация
6	Приложения DNN	Задачи обнаружения, распознавания и предсказания сигналов

Основная литература I

- ① Винер Н. Кибернетика, или Управление и связь в животном и машине. 2-е изд. М.: Советское радио, 1968.
- ② Новиков Д.А. Кибернетика: Навигатор. История кибернетики, современное состояние, перспективы развития. М.: ЛЕНАНД, 2016.
- ③ Табор М. Хаос и интегрируемость в нелинейной динамике. М.: Едиториал УРСС, 2001.
- ④ Кострикин А.И. Введение в алгебру. Часть 2. Линейная алгебра. МЦНМО, 2009.
- ⑤ Измайлов А., Солодов М. Численные методы оптимизации. М.: Физматлит, 2008.
- ⑥ Гнеденко Б.В. Хинчин А.Я. Элементарное введение в теорию вероятностей. М.: Наука, 1970.
- ⑦ Ивченко Г.И., Медведев Ю.И. Введение в математическую статистику. М.: ЛКИ, 2010.
- ⑧ Шипунов А.Б. и др. Наглядная статистика. Используем R! ДМК Пресс, 2012.

Основная литература II

- ⑨ Рашка С. Python и машинное обучение. ДМК Пресс, 2017.
- ⑩ Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. ДМК Пресс, 2017.
- ⑪ Макаренко А.В. Искусственные нейронные сети. / Теория управления (дополнительные главы). М.: ЛЕНАНД, ИПУ РАН, 2019.
- ⑫ Макаренко А.В. Интеллектуальное управление. Введение. / Теория управления (дополнительные главы). М.: ЛЕНАНД, ИПУ РАН, 2019.
- ⑬ Оппенгейм А., Шафер Р. Цифровая обработка сигналов. Изд. 2-е, испр. М.: Техносфера, 2007.
- ⑭ Бендат Дж., Пирсол А. Прикладной анализ случайных данных. М.: Мир, 1989.
- ⑮ Короновский А.А., Храмов А.Е. Непрерывный вейвлетный анализ и его приложения. М.: Физматлит, 2003.

Дополнительная литература

- ① Красовский А.А. (ред.). Справочник по теории автоматического управления. М.: Наука, 1987.
- ② Калман Р., Фалб П., Арbib М. Очерки по математической теории систем. М.: Едиториал УРСС, 2004.
- ③ Гукенхеймер Дж. Нелинейные колебания, динамические системы и бифуркации векторных полей. М.: ИКИ, 2002.
- ④ Кузнецов А.П., Савин А.В., Тюрюкина Л.В. Введение в физику нелинейных отображений. Саратов: Научная книга, 2010.
- ⑤ Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006.
- ⑥ Богданович В. А., Вострецов А. Г. Теория устойчивого обнаружения, различения и оценивания сигналов. 2-е изд., испр. М.: Физматлит, 2004.
- ⑦ Гилл Ф., Мюррэй У. Численные методы условной оптимизации. М.: Мир, 1977.
- ⑧ Кобзарь А.И. Прикладная математическая статистика. М.: Физматлит, 2006.

Критерии формирования оценки

Количество баллов, получаемых слушателем на дифференцированном зачёте, определяется следующими компонентами:

- ❶ x_1 – Посещение лекций
- ❷ x_2 – Посещение семинаров
- ❸ x_3 – Активность на семинарах
- ❹ x_4 – Выполнение домашних заданий
- ❺ x_5 – Уровень владения теоретическим материалом
- ❻ x_6 – Уровень владения практическими навыками
- ❼ x_7 – Выполнение квалификационного задания

Условие получения «автомата»

Слушатель получает автоматический зачёт (отлично) в случае успешного решения двух задач:

Условие получения «автомата»

Слушатель получает автоматический зачёт (отлично) в случае успешного решения двух задач:

Задача 1

Необходимо разработать, обучить и представить предобученную глубокую нейронную сеть принимающую на вход произвольный сигнал, заданный в виде дискретной последовательности $\{s_k\}_{k=1}^{1024}$, $s \in \mathbb{R}$, $t_{k+1} - t_k = \tau$, и выдающую на выходе оценку power spectral density (dB) в виде дискретной последовательности $\{g_k\}_{k=0}^{512}$. Обучающее и тестовое множества генерируются самостоятельно. Решение реализуется на фреймворке Keras (Python 3.x, TF) и проверяется на валидационном наборе лектора.

Критерий прохождения теста: как минимум качественное совпадение (по форме спектра) оценок power spectral density с выхода нейросети и после Фурье-преобразования.

Условие получения «автомата»

Слушатель получает автоматический зачёт (отлично) в случае успешного решения двух задач:

Задача 2

Необходимо разработать, обучить и представить предобученную глубокую нейронную сеть принимающую на вход произвольный сигнал, заданный в виде дискретной последовательности $\{s_k\}_{k=1}^{1024}$, $s \in \mathbb{R}$, $t_{k+1} - t_k = \tau$, и выдающую на выходе оценку показателя Ляпунова в виде числа $\Lambda \in \mathbb{R}$. Обучающее и тестовое множества предоставляются. Решение реализуется на фреймворке Keras (Python 3.x, TF) и проверяется на валидационном наборе лектора.

Критерий прохождения теста: получить качество превосходящее референсное решение, результат оценивается по перцентилям мер MPE (mean percentile error) и MAPE (mean absolute percentile error) на валидационном наборе (для хаотических траекторий с $\Lambda > 0$).

%	5	25	50	75	95
<i>MPE</i>	−21.75	−6.83	0.83	6.16	13.13
<i>MAPE</i>	0.61	3.12	6.39	10.81	45.00

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Определение понятия

КИБЕРНЕТИКА – наука об общих закономерностях получения, хранения, преобразования и передачи информации в сложных управляющих системах, будь то машины, живые организмы или общество.

Определение понятия

КИБЕРНЕТИКА – наука об общих закономерностях получения, хранения, преобразования и передачи информации в сложных управляющих системах, будь то машины, живые организмы или общество.

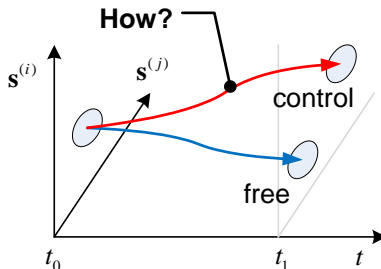
ТЕОРИЯ УПРАВЛЕНИЯ – наука о принципах и методах управления различными системами, процессами и объектами. Теоретической базой теории управления являются кибернетика и теория информации.

Определение понятия

КИБЕРНЕТИКА – наука об общих закономерностях получения, хранения, преобразования и передачи информации в сложных управляющих системах, будь то машины, живые организмы или общество.

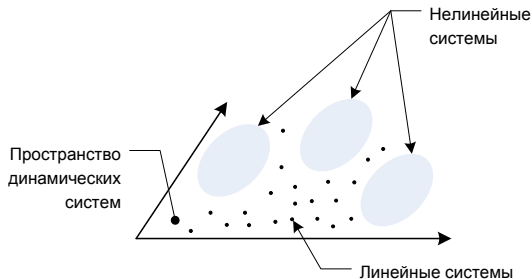
ТЕОРИЯ УПРАВЛЕНИЯ – наука о принципах и методах управления различными системами, процессами и объектами. Теоретической базой теории управления являются кибернетика и теория информации.

УПРАВЛЕНИЕ – целенаправленный перевод (переход) системы Σ посредством управляющего воздействия U из одного состояния S_b в другое – требуемое S_e (определённое целями управления W_u).



Определение понятия

НЕЛИНЕЙНАЯ ДИНАМИКА – междисциплинарная наука, в которой изучаются свойства нелинейных динамических систем. Нелинейная динамика использует для описания систем нелинейные модели, обычно описываемые дифференциальными уравнениями или дискретными отображениями.



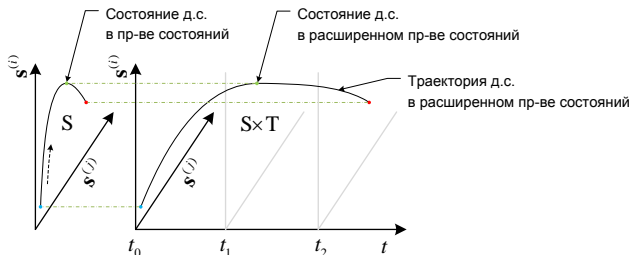
$$\frac{d}{dt} \mathbf{s} \equiv \dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{p}),$$

$$\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{p}).$$

Определение понятия

НЕЛИНЕЙНАЯ ДИНАМИКА – междисциплинарная наука, в которой изучаются свойства нелинейных динамических систем. Нелинейная динамика использует для описания систем нелинейные модели, обычно описываемые дифференциальными уравнениями или дискретными отображениями.

ДИНАМИЧЕСКАЯ СИСТЕМА – множество связанных элементов, для которого задана функциональная зависимость между временем и состоянием – положением в пространстве состояний каждого элемента системы. Данная математическая абстракция позволяет описывать и изучать эволюцию систем во времени. Свойства и характеристики нелинейных д. систем зависят от их состояния.

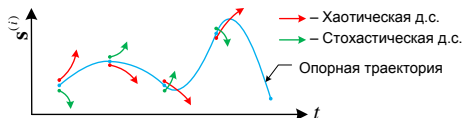


Определение понятия

НЕЛИНЕЙНАЯ ДИНАМИКА – междисциплинарная наука, в которой изучаются свойства нелинейных динамических систем. Нелинейная динамика использует для описания систем нелинейные модели, обычно описываемые дифференциальными уравнениями или дискретными отображениями.

ДИНАМИЧЕСКАЯ СИСТЕМА – множество связанных элементов, для которого задана функциональная зависимость между временем и состоянием – положением в пространстве состояний каждого элемента системы. Данная математическая абстракция позволяет описывать и изучать эволюцию систем во времени. Свойства и характеристики нелинейных д. систем зависят от их состояния.

ДИНАМИЧЕСКИЙ ХАОС – явление в теории динамических систем, при котором поведение нелинейной системы выглядит случайным, несмотря на то, что оно определяется детерминированными правилами (законами).



Определение понятия

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

Определение понятия

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

ИСКУССТВЕННАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА – это аппаратно-программный комплекс, способный решать творческие задачи, традиционно считающиеся прерогативой человека.

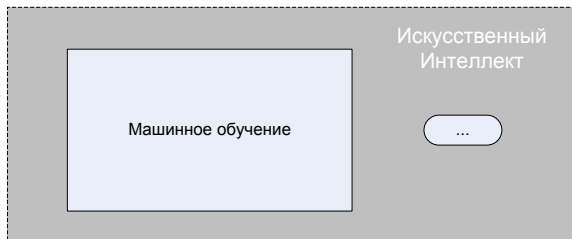
Определение понятия

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

ИСКУССТВЕННАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА – это аппаратно-программный комплекс, способный решать творческие задачи, традиционно считающиеся прерогативой человека.

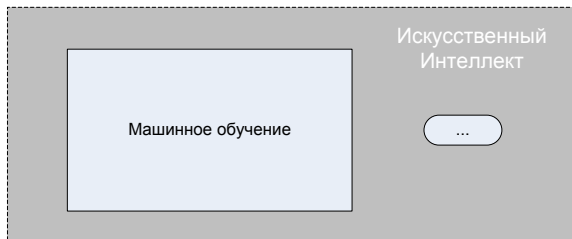
ТВОРЧЕСТВО – процесс деятельности, создающий качественно новые материальные и духовные ценности или итог создания объективно нового. Основным критерий, отличающий творчество от изготовления (производства) – уникальность его результата. Результат творчества невозможно прямо вывести из начальных условий. Именно этот факт придаёт продуктам творчества дополнительную ценность в сравнении с продуктами производства.

Определение понятия



МАШИННОЕ ОБУЧЕНИЕ – обширный (центральный) подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Определение понятия



МАШИННОЕ ОБУЧЕНИЕ – обширный (центральный) подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Различают два типа обучения машин:

- Дедуктивное обучение – предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний (область экспертных систем).
- Индуктивное обучение – (обучение по прецедентам) – основано на выявлении общих закономерностей по частным эмпирическим (экспериментальным) данным.

Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Наука, 1974.

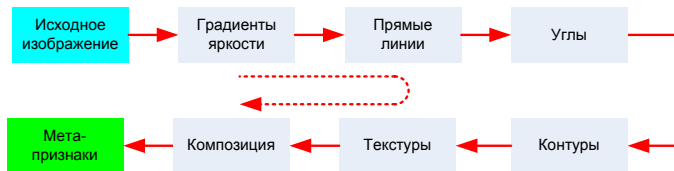
Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

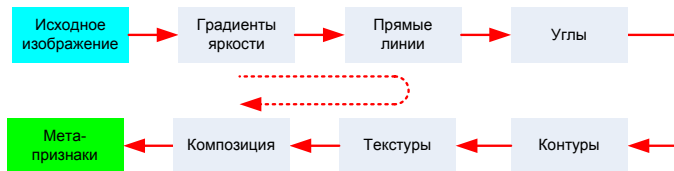
Пример (распознавание изображений):



Определение понятия

ГЛУБОКОЕ ОБУЧЕНИЕ – набор алгоритмов машинного обучения, которые пытаются моделировать иерархические абстракции в данных, используя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров).

Пример (распознавание изображений):



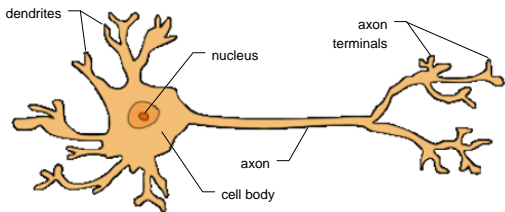
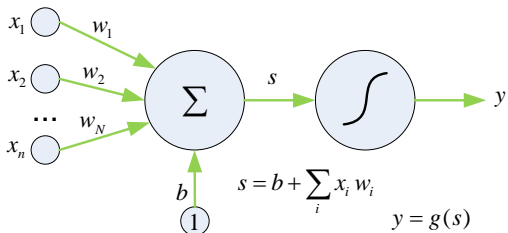
Уникальная особенность глубокого обучения – работа с исходными данными (низкоуровневыми признаками) и самостоятельное извлечение (формирование) признаков описания объектов. Т.е. речь идёт о метаобучении – программа самостоятельно учится как лучше ей учиться.

Определение понятия

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.

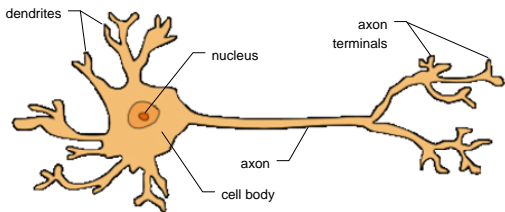
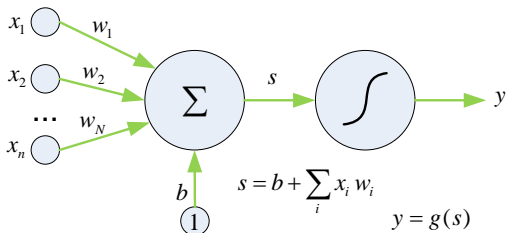
Определение понятия

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.



Определение понятия

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ – это многослойные искусственные нейронные сети, с числом внутренних (скрытых) слоёв более одного.













Примечание: на данный момент глубокие нейронные сети – это основная парадигма Deep Learning.

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python**
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Почему Python?

IEEE 2016 Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016>

Макаренко А.В. Комплексный анализ данных и машинное обучение: 8 причин для миграции с Wolfram Mathematica на Python/R, www.rdcn.ru [Мнение, 2016].

История создания



- **Автор языка:** Гвидо ван Россум (Guido van Rossum), голландец. В проекте имеет статус «великодушного пожизненного диктатора».
- **Мотивация названия:** автор назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона».
- **Дата первого релиза:** 20 февраля 1991 г.
- **Эталонная реализация:** CPython.
- **Лицензия:** свободная лицензия Python Software Foundation License, позволяет использовать язык без ограничений в любых приложениях, включая проприетарные.

Основные свойства языка

Парадигмы программирования:

- Императивная (процедурный, структурный, модульный подходы).
- Объектно-ориентированная.
- Функциональная.

Особенности языка:

- Универсальный, высокоуровневый со встроенными высокоуровневыми структурами данных.
- Интерпретируемый (поддерживает REPL среду).
- Динамическая типизация, автоматическое управление памятью.
- Полная интроспекция, механизм обработки исключений.
- Синтаксис ядра минималистичен, стандартная библиотека весьма ёмкая.
- Код организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).
- Интегрируется с другими языками (C/C++, R, Java, ...).

Два аспекта языка

Две несовместимые ветки:

- Python 2.7.x. (поддержка завершена)
- Python 3.x.

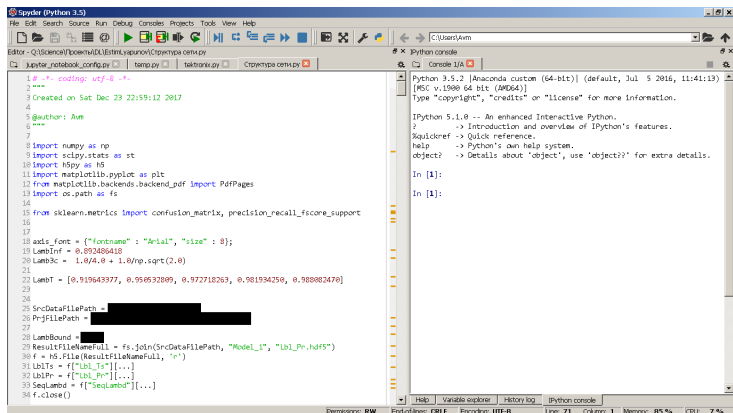
Данный курс ориентирован на версию Python не ниже 3.6.

Рекомендации по оформлению кода PEP 8:

- 1 шаг отступа – 4 пробела.
- Максимальная длина строки – 79 символов.
- Функции – `f_name`.
- Переменные – `var_name`.
- Константы – `CONST_NAME`.
- ... [PythonWorld.ru](https://pythonworld.ru) [PEP 8]

На практике различные команды устанавливают свои требования, и мы не исключение.

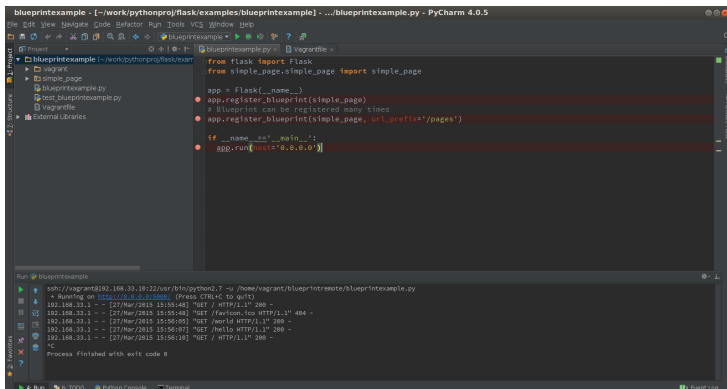
IDE Spyder



Spyder – свободная и кроссплатформенная интерактивная IDE для научных расчётов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части.

Web-сайт документации: [pythonhosted.org \[spyder\]](http://pythonhosted.org/spyder)

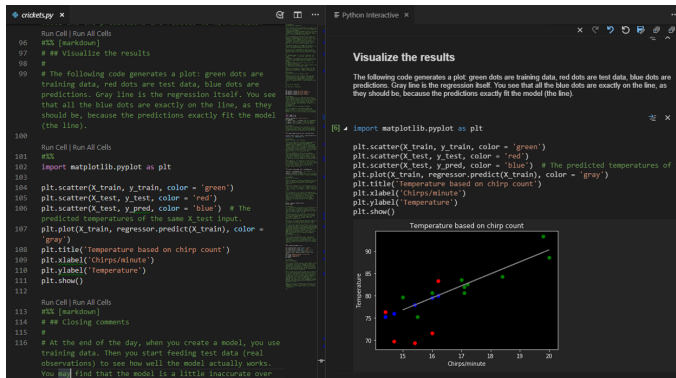
IDE PyCharm



PyCharm – профессиональная интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов. PyCharm разработана компанией JetBrains на основе системы IntelliJ IDEA.

Web-сайт: [jetbrains.com \[pycharm\]](https://jetbrains.com/pycharm)

IDE VS Code



VS Code – профессиональная интегрированная среда разработки для различных языков программирования, в том числе Python. Предоставляет средства для анализа кода, удалённый запуск, REPL, нативную поддержку блокнотов Jupyter, инструмент для запуска юнит-тестов, и т.п. VS Code разработан компанией Microsoft.

Web-сайт: [Visual Studio Code](https://code.visualstudio.com/)

Anaconda

ANACONDA – дистрибутив языка Python, включающий в себя набор библиотек для научных и инженерных расчётов и визуализации, менеджер пакетов *conda*, интерактивную оболочку *Jupyter*.

Web-сайт: [Anaconda.com](https://anaconda.com)

Задание для слушателей:

- Скачать дистрибутив Anaconda3-5.3.0 для версии Python 3.6.
- Установить через менеджер пакетов *conda* дополнительный менеджер *pip*.
- Произвести обновление библиотек: *numpy*, *scipy*, *sklearn*, *pandas*, *matplotlib*.
- Произвести обновление среды программирования *spyder*.
- Приступить к изучению IDE *Spyder* и синтаксиса языка Python.

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python**
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Общие положения

Jupyter Notebook – интерактивная оболочка для ряда языков программирования (Julia, Python, R), позволяющая объединить код, текст, изображения, графики, в один документ и распространять его для других пользователей с сохранением возможности интерактивных «перевычислений».

Общие положения

Jupyter Notebook – интерактивная оболочка для ряда языков программирования (Julia, Python, R), позволяющая объединить код, текст, изображения, графики, в один документ и распространять его для других пользователей с сохранением возможности интерактивных «перевычислений».

Jupyter Notebook является развитием *IPython Notebook*, поддерживает подмножество Markdown – для форматирования текста и LaTeX – для вывода математических формул. Основной элемент это Ячейка (допускает независимое «перевычисление»), их совокупность составляет документ.

Примечание. На подобном принципе отображения информации в виде последовательности взаимосвязанных ячеек построен также Wolfram Mathematica Notebook.

Общие положения

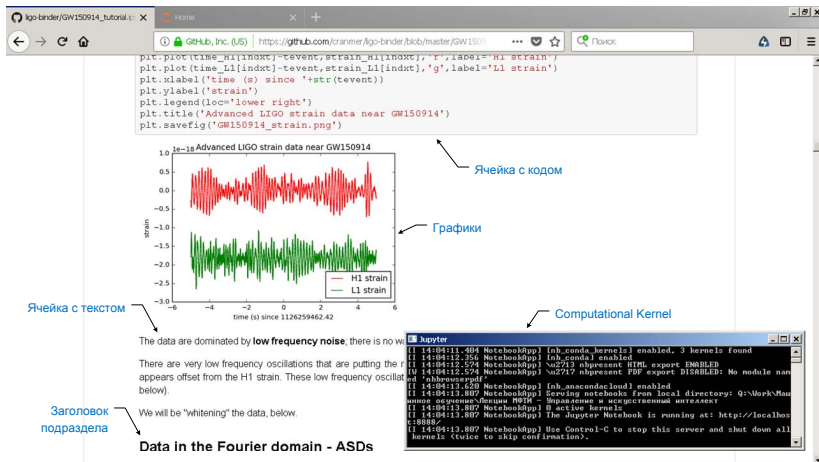
Jupyter Notebook – интерактивная оболочка для ряда языков программирования (Julia, Python, R), позволяющая объединить код, текст, изображения, графики, в один документ и распространять его для других пользователей с сохранением возможности интерактивных «перевычислений».

Jupyter Notebook является развитием *IPython Notebook*, поддерживает подмножество Markdown – для форматирования текста и LaTeX – для вывода математических формул. Основным элементом это Ячейка (допускает независимое «перевычисление»), их совокупность составляет документ.

Примечание. На подобном принципе отображения информации в виде последовательности взаимосвязанных ячеек построен также Wolfram Mathematica Notebook.

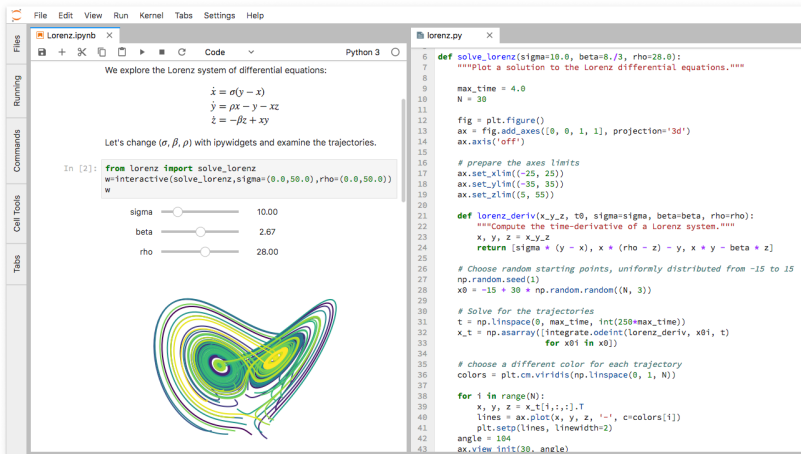
Jupyter Notebook построен по клиент-серверной архитектуре: документ редактируется и отображается в web браузере, обработка ведётся в вычислительном ядре, которое может быть как локальным, так и удалённым.

Пример рабочего окна



Пример: GW150914_tutorial

JupyterLab

Документация: [JupyterLab Documentation](https://jupyterlab.readthedocs.io/en/latest/)

Дополнительные моменты

- Есть возможность выбора web браузера, через редактирование конфигурационного файла `jupyter_notebook_config.py`

```
import webbrowser  
webbrowser.register('firefox', None, webbrowser.GenericBrowser('firefox.exe'))  
c.NotebookApp.browser = 'firefox'
```

- Есть возможность задания рабочей директории для файлов проекта через редактирование ярлыка (пример для MS Win 7):

Объект: "C:/Anaconda3/Scripts/jupyter.exe notebook"
Рабочая папка: "G:/Projects/DL/Test_1"

- Для Jupyter Notebook доступны расширения [Jupyter notebook extensions](#)
- Расширение [RISE](#) – презентации в Jupyter Notebook. См. [пример](#).
- Модуль [nbconvert](#) – преобразование Jupyter Notebook в PDF (отчёты), HTML (посты).

Общие положения

NumPy – это библиотека с открытым исходным кодом для высокоэффективных операций («cycle free») и математических вычислений над многомерными массивами (объект `ndarray`). Дополнительно поддерживаются: файловый ввод-вывод, вызов C/C++ функций.

Документация: [NumPy Manual](#).

Соглашение об импорте: `import numpy as np`.

Особенности:

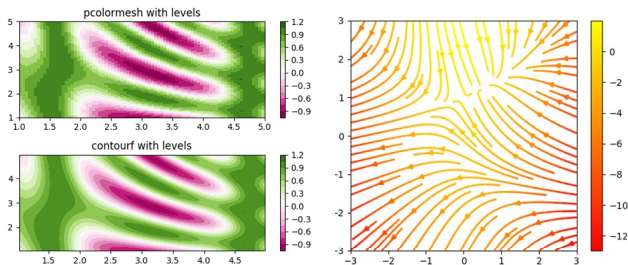
- В отличие от нативных списков Python, массивы NumPy имеют фиксированный (выровненный) размер, элементы массива имеют фиксированный тип.
- Основная парадигма – индексация и слайсинг: `x[:, 1]`, `x[:-1, :]`.
- Поддерживается конвейер (pipeline):
`x = np.arange(9).reshape(3, 3).sum(axis = 1)`.

Общие положения

Matplotlib – это библиотека для визуализации данных посредством построения 2D графиков. При построении графиков используется объектно-ориентированная нотация. Получаемые изображения являются векторными и могут быть экспортированы в ряд форматов (SVG, EPS, PDF, TIFF, PNG, и т.д.).

Документация: [Matplotlib Overview](#).

Соглашение об импорте: `import matplotlib.pyplot as plt`.



Источник: [Matplotlib Gallery](#)

Общие положения

SciPy – это библиотека с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов. Построена по модульному принципу. Основная структура данных – массив **ndarray**. Имеет развитые возможности ввода-вывода данных в различные форматы файлов.

Документация: [SciPy Reference Guide](#).

Соглашение об импорте: `import scipy as sc`.

Основные модули, востребованные в программе курса:

- **fftpack** – Fourier Transforms.
- **signal** – Signal Processing.
- **linalg** – Linear Algebra.
- **stats** – Statistics.
- **io** – File IO.

Общие положения

Pandas – библиотека для обработки и анализа данных, функционирует поверх NumPy и предоставляет две специализированные структуры данных верхнего уровня: **Series** – 1D временные ряды и **DataFrame** – 2D таблицы (аналог **data.frame** из языка R). Основное назначение библиотеки: индексация (в том числе иерархическая) и манипулирование (переформатирование, вставка, удаление, выборка, срез и т.п.) многомерными массивами данных. Имеет развитые возможности ввода-вывода данных в различные форматы файлов и SQL СУБД.

Документация: [Pandas documentation](#).

Соглашение об импорте: `import pandas as pd`.

Особенности:

- В отличие от массивов NumPy **DataFrame** допускает для столбцов разнородный тип.
- Доступ к ячейке **DataFrame** через именованя:
`A["name_col"].loc("name_row")`.
- Для индексов поддерживаются временные метки с дискретой в 1 нс (тип данных NumPy `datetime64`).

Общие положения

Scikit-learn – это библиотека алгоритмов машинного обучения и интеллектуального анализа данных, функционирует поверх NumPy и SciPy. Построена по модульному принципу. Основная структура данных – массив **ndarray**. Имеет развитые возможности ввода-вывода данных в различные форматы файлов.

Документация: [User Guide](#).

Соглашение об импорте: `import sklearn as sk`.

Основные модули, востребованные в программе курса:

- **datasets** – Embedded Dataset.
- **metrics** – Model Evaluation.
- **linear_model**, **naive_bayes**, **neighbors**, **svm**, **tree** – Sup. Alg.
- **cluster** – Unsupervised Clustering Algorithms.
- **decomposition** – Matrix Decomposition.
- **manifold** – Manifold Learning.

Общие положения

Эффективная работа в экосистеме Python, в части интеллектуального анализа данных и машинного обучения, предполагает также хорошее знание следующих дополнительных модулей:

- H5Py – ввод-вывод данных в файлы формата [HDF5](#). [Документация](#).
Соглашение об импорте: `import h5py as h5`.
- SQLite – простая и эффективная SQL СУБД. [Документация](#).
Соглашение об импорте: `import sqlite as sq`.
- OpenCV – мощная библиотека машинного зрения. [Документация](#).
Соглашение об импорте: `import cv2 as cv`.

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

Краткий перечень датасетов

MNIST – коллекция рукописных цифр (размер изображения 28x28 пикселей) состоит из тренировочного (60K образцов) и тестового (10K образцов) наборов. [Web-сайт](#).

Fashion-MNIST – коллекция предметов гардероба, аналог MNIST (размер изображения 28x28 пикселей) состоит из тренировочного (60K образцов) и тестового (10K образцов) наборов. [Web-сайт](#).

AudioSet – коллекция вручную размеченных YouTube видеороликов (2.1 млн.) с выделением временных интервалов и звуковых событий (527 классов). Всего 5.8K часов аннотированного аудио-контента. [Web-сайт](#).

CSTR VCTK Corpus – коллекция речевых данных, произнесённых 109 носителями английского языка с различными акцентами. Каждый из дикторов читает около 400 предложений. [Web-сайт](#).

HolStep – коллекция текстовых фрагментов (2013046 тренировочные, 196030 тестовые) предназначенных для разработки алгоритмов машинного обучения направленных на доказательство формальных утверждений (теорем). [Web-сайт](#).

Поисковик Google по датасетам: [Dataset Search](#)

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования**
- 9 Заключение

Обозначения и соглашения – объекты

скаляр: a – число, $\dim a = 0$.

вектор: \mathbf{a} – одномерный массив, $\dim \mathbf{a} = N$.

единичный вектор: $\mathbf{1}$ – одномерный массив, $\dim \mathbf{1} = N$.

вектор-столбец, - строка: $\dim \mathbf{a}_\parallel = N \times 1$, $\dim \mathbf{a}^\parallel = 1 \times N$,

$$\mathbf{a}_\parallel = \begin{bmatrix} a_1 \\ a_2 \\ \dots \end{bmatrix}, \quad \mathbf{a}^\parallel = [a_1 \quad a_2 \quad \dots].$$

матрица: \mathbf{A} – двумерный массив, $\dim \mathbf{A} = N \times M$.

единичная матрица: \mathbf{E} – двумерный массив, $\dim \mathbf{E} = N \times N$,

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} = \text{diag } \mathbf{1}.$$

«тензор»: \mathbf{A} – N -мерный массив, $\dim \mathbf{A} = N_1 \times N_2 \times \dots$

множество, пространство: S .

Обозначения и соглашения – операции

сложение/вычитание: $\mathbf{a} \pm \mathbf{b}$, $\mathbf{A} \pm \mathbf{B}$.

покомпонентное умножение: $\mathbf{a} * \mathbf{b}$, $\mathbf{A} * \mathbf{B}$.

скалярное умножение: $\mathbf{a} \cdot \mathbf{b}$.

матричное умножение: $\mathbf{A} \mathbf{B}$, $\mathbf{a} \mathbf{b}$.

внешнее умножение: $\mathbf{a} \otimes \mathbf{b}$, $\mathbf{A} \otimes \mathbf{B}$.

декартово произведение $S \times T$.

транспонирование: \mathbf{a}^\top , \mathbf{B}^\top .

комплексное сопряжение: \mathbf{a}^* , \mathbf{B}^* .

эрмитово сопряжение: \mathbf{a}^{\dagger} , \mathbf{B}^{\dagger} .

обратный вектор, матрица: \mathbf{a}^{-1} , \mathbf{B}^{-1} .

норма вектора, матрицы: $|\mathbf{a}|$, $\|\mathbf{A}\|$.

детерминант матрицы: $|\mathbf{A}|$.

мощность множества: $|\mathbf{V}|$.

Сложение, умножение

сложение/вычитание:

$$\mathbf{a} \pm \mathbf{b} = \begin{bmatrix} a_1 \pm b_1 \\ a_2 \pm a_2 \end{bmatrix}, \quad \mathbf{A} \pm \mathbf{B} = \begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} \end{bmatrix}.$$

покомпонентное умножение:

$$\mathbf{a} * \mathbf{b} = \begin{bmatrix} a_1 b_1 \\ a_2 a_2 \end{bmatrix}, \quad \mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11} b_{11} & a_{12} b_{12} \\ a_{21} b_{21} & a_{22} b_{22} \end{bmatrix}.$$

скалярное умножение:

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2, \quad \mathbf{a}^{\top} \cdot \mathbf{b} = [a_1 b_1 + a_2 b_2].$$

матричное умножение:

$$\mathbf{A} \mathbf{B} = \begin{bmatrix} a_{11} b_{11} + a_{12} b_{21} & a_{11} b_{12} + a_{12} b_{22} \\ a_{21} b_{11} + a_{22} b_{21} & a_{21} b_{12} + a_{22} b_{22} \end{bmatrix}, \quad \mathbf{A} \mathbf{B} \neq \mathbf{B} \mathbf{A}.$$

внешнее умножение:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 b_1 & a_1 b_2 \\ a_2 b_1 & a_2 b_2 \end{bmatrix}.$$

Транспонирование, комплексное сопряжение, обращение

транспонирование:

$$\mathbf{a}_1^\top = [a_1 \quad a_2], \quad \mathbf{A}^\top = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}.$$

комплексное сопряжение:

$$\begin{bmatrix} a_{11} + i b_{11} & a_{12} + i b_{12} \\ a_{21} + i b_{21} & a_{22} + i b_{22} \end{bmatrix}^* = \begin{bmatrix} a_{11} - i b_{11} & a_{12} - i b_{12} \\ a_{21} - i b_{21} & a_{22} - i b_{22} \end{bmatrix}.$$

эрмитово сопряжение:

$$\begin{bmatrix} a_{11} + i b_{11} & a_{12} + i b_{12} \\ a_{21} + i b_{21} & a_{22} + i b_{22} \end{bmatrix}^\dagger = \begin{bmatrix} a_{11} - i b_{11} & a_{21} - i b_{21} \\ a_{12} - i b_{12} & a_{22} - i b_{22} \end{bmatrix}.$$

симметрические, ортогональные, эрмитовы и унитарные матрицы:

$$\mathbf{A} = \mathbf{A}^\top, \quad \mathbf{B}^\top \mathbf{B} = \mathbf{E}, \quad \mathbf{C} = \mathbf{C}^\dagger, \quad \mathbf{C}^\dagger \mathbf{C} = \mathbf{E}.$$

обратный вектор, матрица:

$$\mathbf{a}^{-1} \cdot \mathbf{a} = 1, \quad \mathbf{A}^{-1} \mathbf{A} = \mathbf{E}.$$

Собственные значения, базис, SVD-разложение, нормы

собственное значение:

$$\mathbf{A} \mathbf{x} = \lambda \mathbf{x}.$$

ортонормированный набор базисных векторов:

$$\text{если } \mathbf{A}^{-1} = \mathbf{A}^\dagger,$$

тогда столбцы (или строки) матрицы \mathbf{A} образуют базис.

SVD-декомпозиция:

$$\text{если } \mathbf{B} = \mathbf{U} \mathbf{S} \mathbf{V}^\dagger, \quad \dim \mathbf{B} = M \times N,$$

тогда унитарные матрицы \mathbf{U} ($\dim \mathbf{U} = M \times M$) и \mathbf{V} ($\dim \mathbf{V} = N \times N$) содержат левые и правые сингулярные вектора \mathbf{B} , соответственно, а $\text{diag } \mathbf{S}$ – сингулярные числа \mathbf{B} . SVD-разложение имеет широкую применимость в задачах анализа данных и машинного обучения.

нормы векторов и матриц:

$$L_1, \quad L_2, \quad L_\infty.$$

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
- 4 Основные дисциплины курса
- 5 Python
- 6 Библиотеки и инструменты Python
- 7 Открытые датасеты
- 8 Векторно-матричные преобразования
- 9 Заключение

На подумать...

Разработка vs Исследования

В науке, если знаете, что получится – значит занимаетесь не тем.

В инженерном деле, если не знаете, что получится – не стоит этого делать.

Ричард Хэмминг.