

УПРАВЛЕНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

ВВОДНАЯ ЛЕКЦИЯ

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
15 февраля 2018 г.
ИПУ РАН, Москва, Россия

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 Заключение

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения
О курсе
- 4 Основные дисциплины курса
Кибернетика
Нелинейная динамика
Искусственный интеллект
Машинное обучение
Глубокое обучение
Глубокие нейронные сети
- 5 Python
Общие положения
Интегрированная среда разработки
Дистрибутив
- 6 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
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 Заключение

Общая информация

- Основатель и руководитель научно исследовательской группы «Конструктивная Кибернетика»
- Заведующий лабораторией 77 «Вычислительная Кибернетика» ИПУ РАН
- Старший научный сотрудник ИПУ РАН
- Кандидат технических наук по специальности «Системный анализ, управление и обработка данных»
- IEEE Signal Processing Society Membership
- IEEE Computational Intelligence Society Membership
- Сертифицированный эксперт-инструктор Wolfram Research Inc

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++, Python, R, Wolfram Language
- Технологии: MPI, OpenMP, Cilk Plus, CUDA
- Фреймворки и библиотеки: Keras, Berkeley Caffe, Google TensorFlow, MS CNTK, Theano, MXNet, Intel MKL, Intel MPP, ...

Outline section

- 1 Предисловие к курсу
- 2 Об авторе
- 3 Общие положения**
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 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 3.5. Дистрибутив Anaconda. Notebook Jupyter. IDE Spyder. Библиотеки Numpy, Matplotlib, Scipy, Pandas, Scikit-learn. Базовые датасеты.
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 мульти-модальные 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 Оппенгейм А., Шафер Р. Цифровая обработка сигналов. Изд. 2-е, испр. М.: Техносфера, 2007.
- ⑫ Бендат Дж., Пирсол А. Прикладной анализ случайных данных. М.: Мир, 1989.
- ⑬ 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.

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, CNTK) 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, CNTK) 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
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 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.

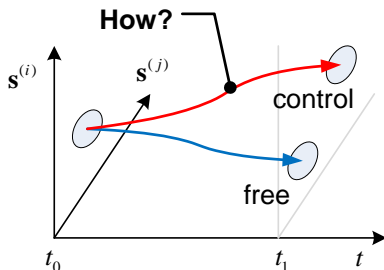
ТЕОРИЯ УПРАВЛЕНИЯ – наука о принципах и методах управления различными системами, процессами и объектами. Теоретической базой теории управления являются кибернетика и теория информации.

Определение понятия

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

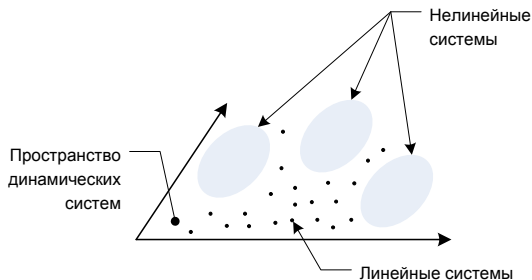
ТЕОРИЯ УПРАВЛЕНИЯ – наука о принципах и методах управления различными системами, процессами и объектами. Теоретической базой теории управления являются кибернетика и теория информации.

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.



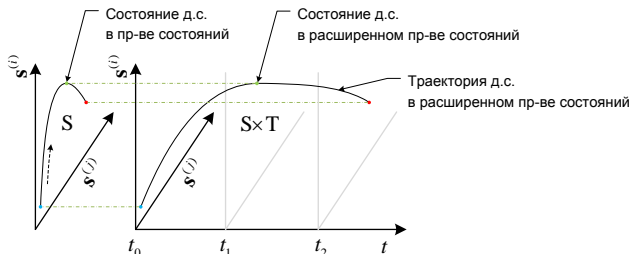
$$\frac{d}{dt} \mathbf{s} \equiv \dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{p}),$$

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

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.

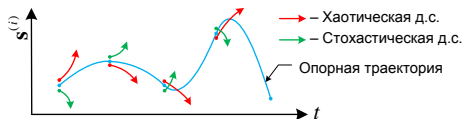


Определение понятия

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.

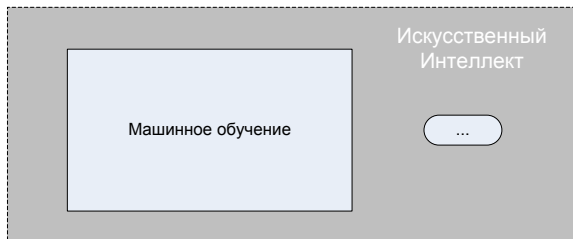
Определение понятия

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.

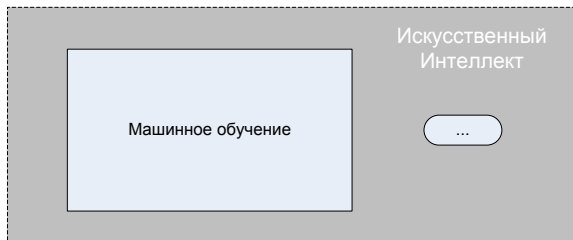
ТВОРЧЕСТВО – процесс деятельности, создающий качественно новые материальные и духовные ценности или итог создания объективно нового. Основным критерий, отличающий творчество от изготовления (производства) – уникальность его результата. Результат творчества невозможно прямо вывести из начальных условий. 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).

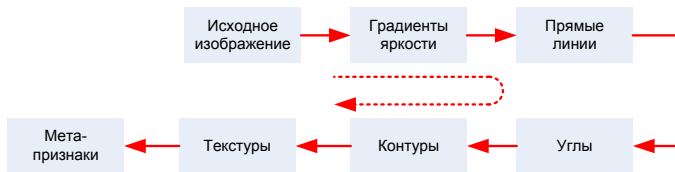
Пример (распознавание изображений):



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):



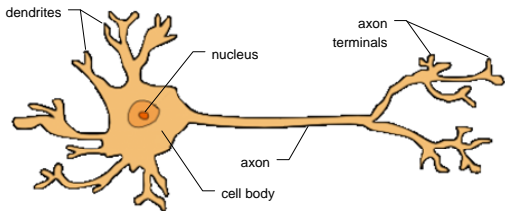
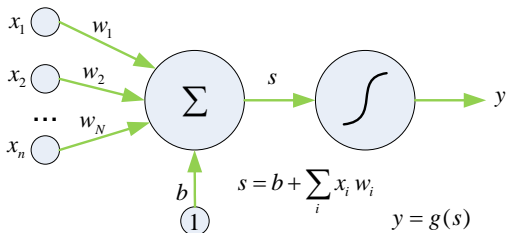
Уникальная особенность глубокого обучения – работа с исходными данными (низкоуровневыми признаками) и самостоятельное извлечение (формирование) признаков описания объектов. Т.е. речь идёт о метаобучении – программа самостоятельно учится как лучше ей учиться.

Определение понятия

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

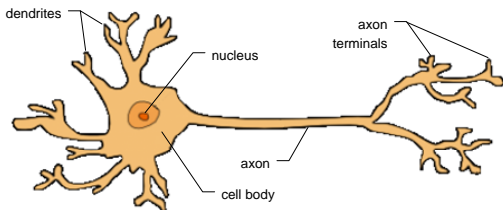
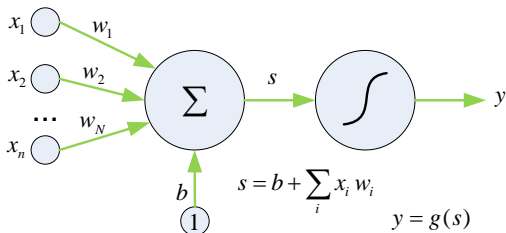
Определение понятия

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
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 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.x.
- Python 3.x.

Данный курс ориентирован на версию Python не ниже 3.5.

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 [PEP 8]

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

IDE Spyder

```

1 0 -*- coding: utf-8 -*-
2
3 Created on Sat Dec 23 22:59:12 2017
4
5 @author: Am
6
7
8 import numpy as np
9 import scipy.stats as st
10 import h5py as h5
11 import matplotlib.pyplot as plt
12 from matplotlib.backends.backend_pdf import PdfPages
13 import os.path as fs
14
15 from sklearn.metrics import confusion_matrix, precision_recall_fscore_support
16
17
18 axis_font = {"fontname": "Arial", "size": 8};
19 LambInf = 0.892486418
20 Lamb3c = 1.0/4.0 + 1.0/np.sqrt(2.0)
21
22 LambT = [0.919643377, 0.950632809, 0.972718263, 0.981934250, 0.988882470]
23
24
25 SrcDataFilePath =
26 PrjFilePath =
27
28 LambBound =
29 ResultFileNameFull = fs.join(SrcDataFilePath, "Model_1", "Lbl_Pr_{}.hdf5")
30 f = h5.File(ResultFileNameFull, 'r')
31 LblTs = f["Lbl_Ts"][...]
32 LblPr = f["Lbl_Pr"][...]
33 SeqLambd = f["SeqLambd"][...]
34 f.close()

```

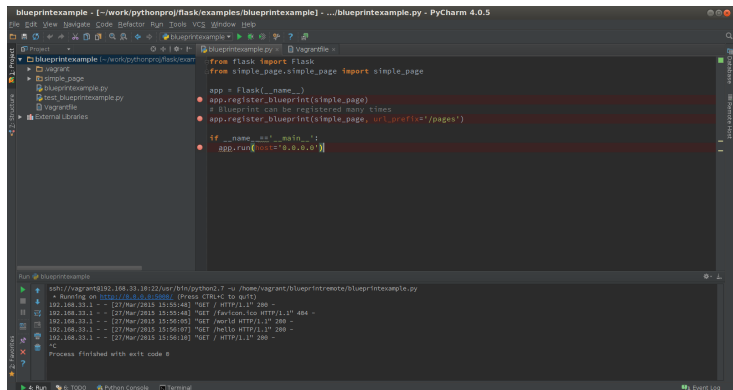
Python 3.5.2 [Anaconda custom (64-bit)] (default, Jul 5 2016, 11:41:13)
 [MSC v.1900 64 bit (AMD64)]
 Type "copyright", "credits" or "license()" for more information.
 IPython 5.1.0 -- An enhanced Interactive Python.
 ? -> Introduction and overview of IPython's features.
 %quickref -> Quick reference.
 help -> Python's own help system.
 object? -> Details about 'object', use 'object??' for extra details.
 In [1]:
 In [1]:

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 71 Column: 1 Memory: 85% CPU: 7%

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 \[pycharm\]](http://jetbrains.com/pycharm)

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 Anaconda3-4.2.0 для версии Python 3.5.
- 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
О курсе
- 4 Основные дисциплины курса
 - Кибернетика
 - Нелинейная динамика
 - Искусственный интеллект
 - Машинное обучение
 - Глубокое обучение
 - Глубокие нейронные сети
- 5 Python
 - Общие положения
 - Интегрированная среда разработки
 - Дистрибутив
- 6 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.