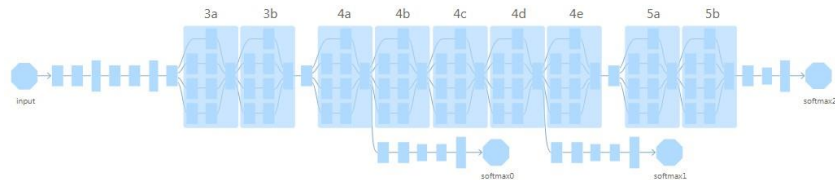




浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE



深度学习应用开发

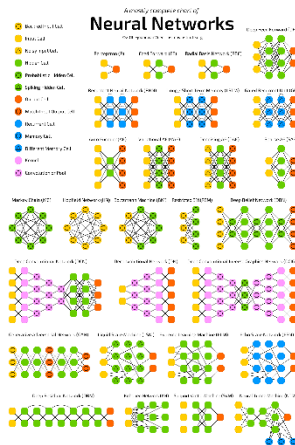
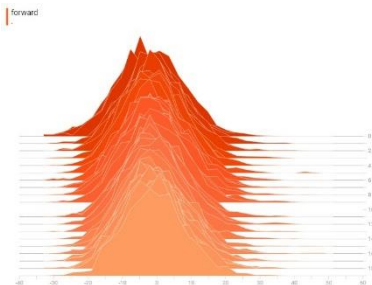
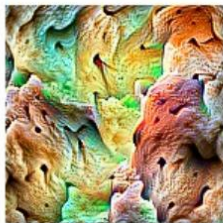
基于TensorFlow的实践

吴明晖 李卓蓉 金苍宏

浙江大学城市学院

计算机与计算科学学院

Dept. of Computer Science
Zhejiang University City College





浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE

简明Python基础



2018年主流十大编程语言





浙江大學城市學院
ZHEJIANG UNIVERSITY CITY COLLEGE



Guido van Rossum

Python是一门面向对象编程语言

由荷兰人Guido van Rossum于1989年发明

Python主要发布了两个大的版本Python2和Python3

Python2和Python3存在一定的区别

将于2020年1月1日起停止对Python2的支持和维护

人生苦短，我用Python



Python基础串讲



print 简单输出方法



print 基础用法



print 函数 - 在终端中输出，**Python 3.x** 需要加 **()**，**Python 2.x** 没有 **()**

print 默认输出是换行的

```
In [2]: print("Hello")  
        print("world!")
```

```
Hello  
world!
```

如果要实现**print**输出不换行，需要指定结尾符 **end=' '**

```
In [17]: print("Hello", end=' ')  
         print("world!")
```

```
Hello world!
```



变量与基本数据类型



变量



每个变量在内存中创建，都包括变量的标识，名称和数据这些信息

每个变量在使用前都必须赋值，赋值号是“=”

```
In [19]: int_var = 3  
float_var = 3.1415926  
str_var = "Hello"  
  
print(int_var, float_var, str_var)
```

```
3 3.1415926 Hello
```



变量的数据类型



Python 中的变量赋值不需要数据类型声明，数据类型根据具体的赋值确定

```
In [20]: print(int_var, type(int_var))  
          print(float_var, type(float_var))  
          print(str_var, type(str_var))
```

```
3 <class 'int'>
```

```
3.1415926 <class 'float'>
```

```
Hello <class 'str'>
```



浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE

标识符



Python的标识符



- 在 **Python** 里，标识符由字母、数字、下划线组成
- 在 **Python** 中，所有标识符可以包括英文、数字以及下划线_，但不能以数字开头

```
In [49]: x_6 = 5
```

```
In [50]: # 错误标识符
```

```
6_x = 5
```

```
File "<ipython-input-50-c6ac8572e4ce>", line 1
```

```
6_x = 5
```

```
^
```

```
SyntaxError: invalid syntax
```



Python的标识符



Python 中的标识符是区分大小写的

In [56]: # 标识符区分大小写

```
x_6 = 5
```

```
X_6 = 6
```

```
print(x_6,X_6)
```

```
5 6
```



Python的标识符



Python3直接支持中文符号，包括标识符名

```
In [53]: #Python3直接支持中文标识符
```

```
变量1 = 8  
print(变量1)
```

8



浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE

保留字



Python的保留字



- 保留字不能用作常数或变数，或任何其他标识符名称
- 所有 **Python** 的关键字只包含小写字母

and	exec	not	assert	finally	or
break	for	pass	class	from	print
continue	global	raise	def	if	return
del	import	try	elif	in	while
else	is	with	except	lambda	yield



Numbers (数字) 数据类型



数字数据类型



数字数据类型用于存储数值

Python支持不同的数字类型：

- `int`（有符号整型），缺省十进制，还可以表示2进制、8进制和16进制；
`Python3` 不再保留长整形，统一为`int`
- `float`（浮点型），可以用科学计数法表示
- `complex`（复数），复数由实数部分和虚数部分构成，可以用 `a + bj` 或者 `complex(a,b)` 表示，复数的实部 `a` 和虚部 `b` 都是浮点型



数字数据类型



0b, 0o, 0x 分别代表2进制、8进制和16进制数

```
In [21]: var1 = 0b10; var2 = 0o10; var3 = 0x10  
print (var1, var2, var3)
```

2 8 16

多条语句可以放在一行，中间用 分号 “;” 隔开



数字数据类型



Python直接支持很长的整数

```
In [22]: var1 = 1234567891012345678901234567890  
  
         print (var1, type(var1))  
  
1234567891012345678901234567890 <class 'int'>
```



数字数据类型



float - 浮点数, 例子: -1.8, 3.5, 1e-5, 6.7e15

```
In [23]: var1 = 1e-5; var2 = 6.7e15; var3 = 6.7e16; var4 = -1.8  
print (var1, var2, var3, var4)
```

```
1e-05 6700000000000000.0 6.7e+16 -1.8
```



数字数据类型



复数由实数部分和虚数部分构成，可以用 $a + bj$ 或者 `complex(a,b)` 表示，复数的实部 a 和虚部 b 都是浮点型

```
In [24]: var1 = 3+5.3j; var2 = complex(3.4e5,7.8)

print (var1, type(var1), var2, type(var2))

(3+5.3j) <class 'complex'> (340000+7.8j) <class 'complex'>
```



布尔类型



布尔类型, **bool** 值: True and False

```
In [25]: i_love_you = True  
         you_love_me = False  
         print(i_love_you, type(i_love_you))  
         print(you_love_me, type(you_love_me))
```

```
True <class 'bool'>  
False <class 'bool'>
```



浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE

注释



注释



单行注释用"#"

```
In [26]: print("这句能打印出来")  
  
# print("这句不能打印出来")
```

这句能打印出来

多行注释用'''或者'''

```
In [27]: print("这句能打印出来")  
  
''' print("这句不能打印出来")  
print("这句也不能打印出来")  
print("这句还不能打印出来")  
'''  
  
print("这句可以打印出来了")
```

这句能打印出来
这句可以打印出来了



基本运算



算术运算



算术运算



算术运算符，包括 +、-、*、/、//，%

以下假设变量： **x=10, y=3**

运算符	描述	实例
+	加：两个对象相加	$x + y$ 输出结果 13
-	减：一个数减去另一个数	$x - y$ 输出结果 7
*	乘：两个数相乘	$x * y$ 输出结果 30
/	除：x除以y	x / y 输出结果 3.3333333333333335
%	取模：返回除法的余数	$x \% y$ 输出结果 1
**	幂：返回x的y次幂	$x ** y$ 为10的3次方， 输出结果 1000
//	取整除：返回商的整数部分（向下取整）	$x // y$ 输出结果 3 , $10 // 3.0$ 输出结果 3.0



比较运算



比较运算



比较运算符，包括 >、<、==、>=、<=，!=
比较运算的结果为 **True** 或者 **False**

以下假设变量： **x=10, y=3**

运算符	描述	实例
==	等于 - 比较对象是否相等	(x == y) 返回 False。
!=	不等于 - 比较两个对象是否不相等	(x != y) 返回 True。
<>	不等于 - 比较两个对象是否不相等	(x <> y) 返回 True。 这个运算符类似 != 。
>	大于 - 返回x是否大于y	(x > y) 返回 False。
<	小于 - 返回x是否小于y。所有比较运算符返回1表示真，返回0表示假。这分别与特殊的变量True和False等价。	(x < y) 返回 True。
>=	大于等于 - 返回x是否大于等于y。	(x >= y) 返回 False。
<=	小于等于 - 返回x是否小于等于y。	(x <= y) 返回 True。



赋值运算



赋值运算



赋值运算符

运算符	描述	实例
=	简单的赋值运算符	$z = x + y$ 将 $x + y$ 的运算结果赋值
+=	加法赋值运算符	$z += x$ 等效于 $z = z + x$
-=	减法赋值运算符	$z -= x$ 等效于 $z = z - x$
*=	乘法赋值运算符	$z *= x$ 等效于 $z = z * x$
/=	除法赋值运算符	$z /= x$ 等效于 $z = z / x$
%=	取模赋值运算符	$z \% = x$ 等效于 $z = z \% x$
**=	幂赋值运算符	$z ** = x$ 等效于 $z = z ** x$
//=	取整除赋值运算符	$z //= x$ 等效于 $z = z // x$



浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE

逻辑运算



逻辑运算符

运算符	逻辑表达式	描述
and	x and y	逻辑“与”：如果 x 为 False，x and y 返回 x，否则它返回 y 的计算值。
or	x or y	逻辑“或”：如果 x 是非 0，它返回 x 的值，否则它返回 y 的计算值。
not	not x	逻辑“非”：如果 x 为 True，返回 False。如果 x 为 False，它返回 True。



逻辑运算



逻辑运算中，非0的值视作True

```
In [75]: # x and y 如果x为False, 则返回False, 否则返回y的计算值  
10 and 20+5
```

Out[75]: 25

逻辑运算中，0视作False

```
In [80]: 0 and 20+5
```

Out[80]: 0

```
In [81]: False and 20+5
```

Out[81]: False



逻辑运算



逻辑运算中，**False**的值就是0

```
In [82]: 0 == False
```

```
Out[82]: True
```

逻辑运算中，**True**的值就是1

```
In [83]: 1 == True
```

```
Out[83]: True
```

逻辑运算中，非0的值只是视作**True**，但并不等于**True**。只有 **1** 等于 **True**

```
In [84]: 5 == True
```

```
Out[84]: False
```



逻辑运算



逻辑‘或’ `or: x or y` 如果`x`为`True`，则返回`True`，否则返回`y`的计算值

```
In [85]: 10 or 20+5
```

```
Out[85]: 10
```

```
In [86]: 0 or 20+5
```

```
Out[86]: 25
```

逻辑‘非’ `not: not x`, 如果`x`为`True`，则返回`False`; 如果`x`为`False`，则返回`True`

```
In [87]: not True
```

```
Out[87]: False
```

```
In [88]: not False
```

```
Out[88]: True
```