1. **基础学习**

**Python 21 元组，列表**

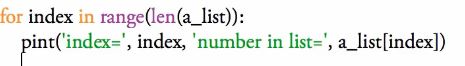
a\_tuple=(12,3,5,15,6)

Another\_tuple=2,4,6,7,8

a\_list=[12,3,67,7,82]

for content in a\_tuple

print(content)



* Python数组中，下标是从0开始的。

**Python 22 列表**

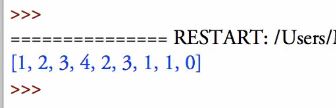
* 增加功能

a=[1,2,3,4,2,3,1,1]

a.append(0)

print(a)

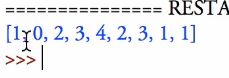
运行结果：



a.insert(1,0)

print(a)

运行结果：

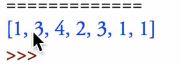


* 减少功能

a.remove(2)

print(a)

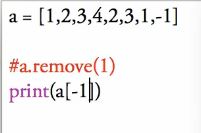
运行结果：



\*减掉了第一次出现的“2”

* **列表索引**

程序：

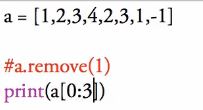


运行结果：

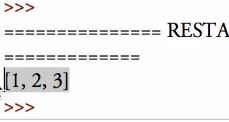


* **多个索引**

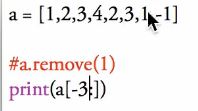
程序：



运行结果：



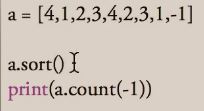
* **倒序索引**



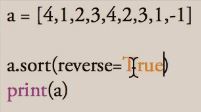
* **下标索引**

**print(a.int(4))**

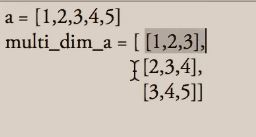
* **排序**

****

* **从大到小排序**

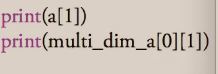
****

**Python 23 多维列表**

****

* **索引**

**程序 ：**

****

**运行结果：**

****

**Python 24 字典（无序容器）**

* **输出字典元素**

**a\_list=[1,2,3,4,5,4]**

**d={‘apple’:1,’pear’:2,’orange’:3}**

**d2={1:’a’,2:’b’}**

**print(d[‘apple’])**

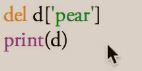
**print(a\_list[0])**

**运行结果：**

****

* **删除字典元素**

**程序：**

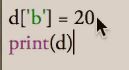
****

**运行结果：**

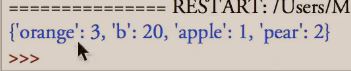
****

* **增加元素**

**程序：**

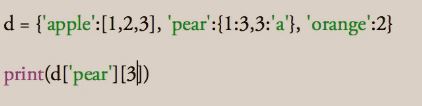
****

**运行结果：**

****

* **复合情况引用**

**程序：**

****

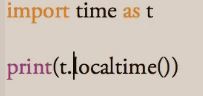
**运行结果：**

****

**Python 25 import载入模块**

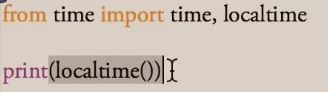
* **模块名称简化**

**程序：**

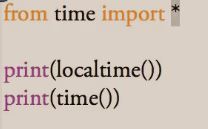
****

* **指定使用模块功能**

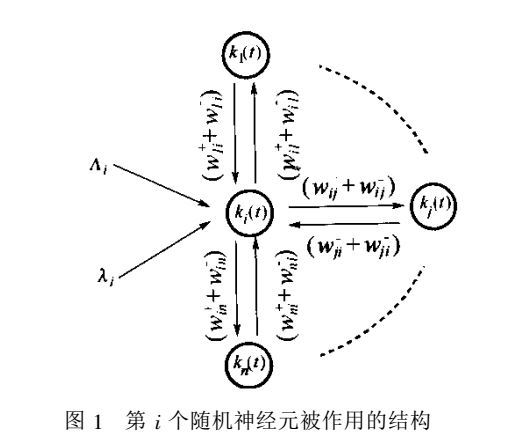
**程序：**

****

* **使用模块所有功能**

****

1. **论文《随机神经网络现状发展综述》**
2. **GNN模型描述**

****

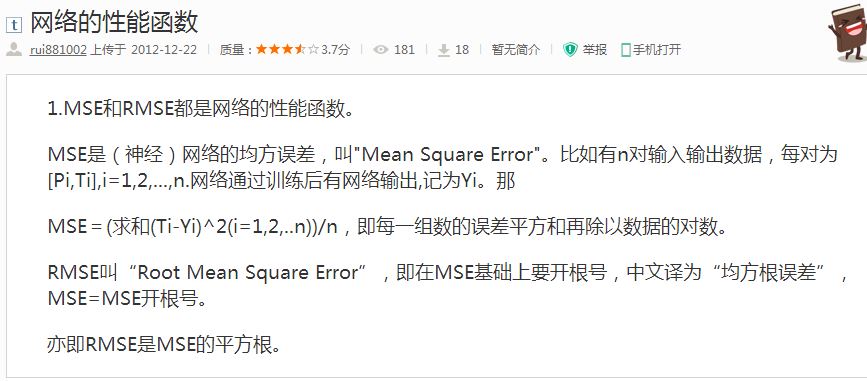
1. **随机网络标准学习算法**

此算法目的是为了得到一个适当的权值矩阵，使网络输出为期望值或者与期望值二次方差最小。

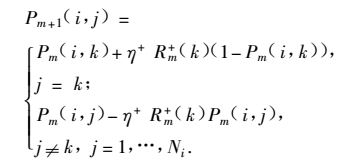
直观结果，使性能函数为最小：

26

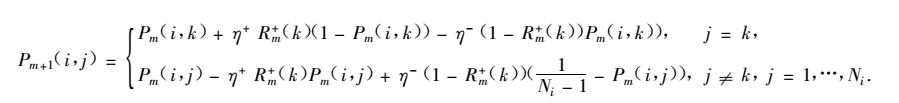
**\*什么是性能函数**

****

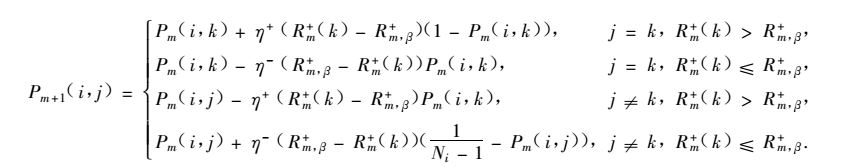
1. **RNN的强化学习算法**
2. **R规则（仅奖励）**

****

1. **L规则（奖励与惩罚并进）**

****

1. **E规则（动态奖惩机制，有遗忘功能）**

****

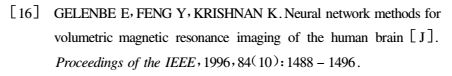
1. **RNN的应用**
2. 人工纹理生成

从彩色纹理图像中提取出纹理特征、信号类别对应色彩的类别，合成一个与原纹理相似的纹理。

【GRNN的权值学习规则？】

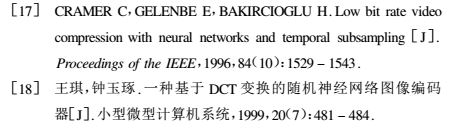
1. 核磁共振图像特征信息提取

文献【16】



1. 图像编码器

文献【17】【18】



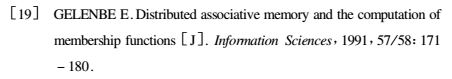
1. 增强图像放大

文献【6】

30

1. 其他运用

文献【19】



1. **其他随机网络**
2. Hopfield网络（单层全反馈）
3. 模拟退火算法（将能量跳出局部最小值的算法）
4. BM网络（双向连接网络）
5. **RNN的研究前景**

**RNN网络已被证明可以作为连续函数的广义函数逼近器。**