


1. 摘要
2. CNN (卷积神经网络)
 - 2.1. 卷积层
 - 2.2. 池化层 (汇聚层)
 - 2.2.1. 非线性激活函数
 - 2.3. 全连接层
 - 2.4. 神经网络的训练与优化
3. 卷积神经网络VS传统神经网络
4. RNN (递归神经网络)
5. LSTM (长短期记忆网络)
6. 自然语言处理
 - 6.1. 词法分析
 - 6.1.1. 分词
 - 6.1.2. 命名实体识别
 - 6.2. 句法分析
 - 6.3. 语义分析
7. Typora常用快捷键
8. Markdown小插曲
 - 8.1. 插入代码块
 - 8.2. 内联HTML代码
 - 8.3. 内联 $LaTeX$ 公式

1. 摘要

本文主要是一些对深度学习的记录，同时在Vscode/Typora上尽可能地多种多样地展示了markdown的基础、进阶功能，包括脚注、公式、引用、任务列表、代码块的行号、emoji表情、目录、高亮、内联HTML代码、内联 $LaTeX$ 公式等，如对markdown语法不是很熟悉的可以食用Typora来写markdown

因为我用的图床是Github上的，所以访问不了GitHub的小伙伴应该是看不到图片的

2. CNN (卷积神经网络)

还可用来进行边界检测等,著名的有手写字体识别模型 LeNet5

2.1. 卷积层

局部连接、窗口滑动¹

- **padding**的目的是为了不至于经过不断过滤器过滤后的输出层收缩过快，防止边缘信息丢失，有vaild和same两种
- 可以有多个**filter**（滤波器），尽量选择偶数，filter不需要我们自己设计，让机器去学习这些参数
- **stride**（步长）默认为1，可以自己设置



输出层
宽度: $\frac{W - F_w + 2P}{S} + 1$

输出层
高度: $\frac{H - F_h + 2P}{S} + 1$

H: 输入层高度
W: 输入层宽度
D: 输入层深度
Fw: 过滤层宽度
Fh: 过滤层高度
P: 填充度
S: 跳跃度

2.2. 池化层（汇聚层）

池化层也可以理解成特征映射

2.2.1. 非线性激活函数

四级

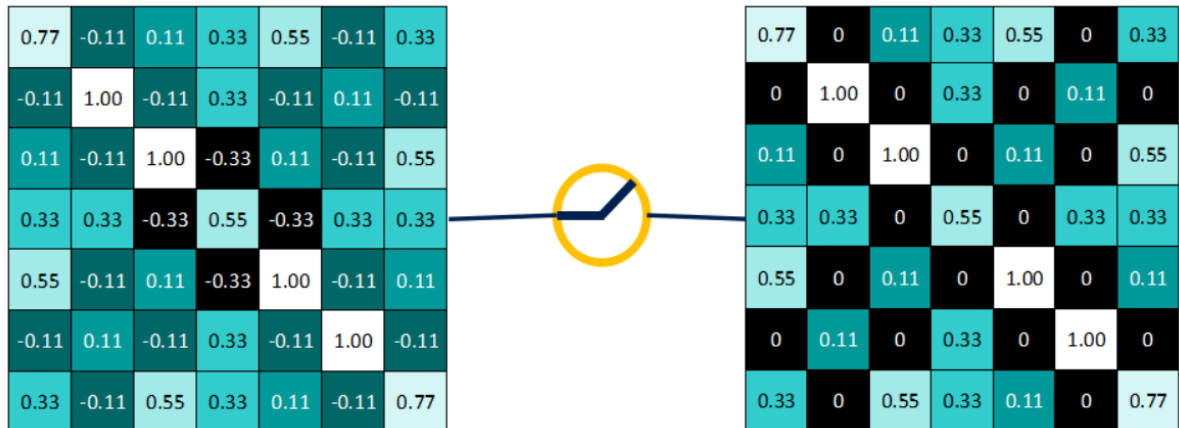
五级

六级

主要参考[教你用css](#)来改编的

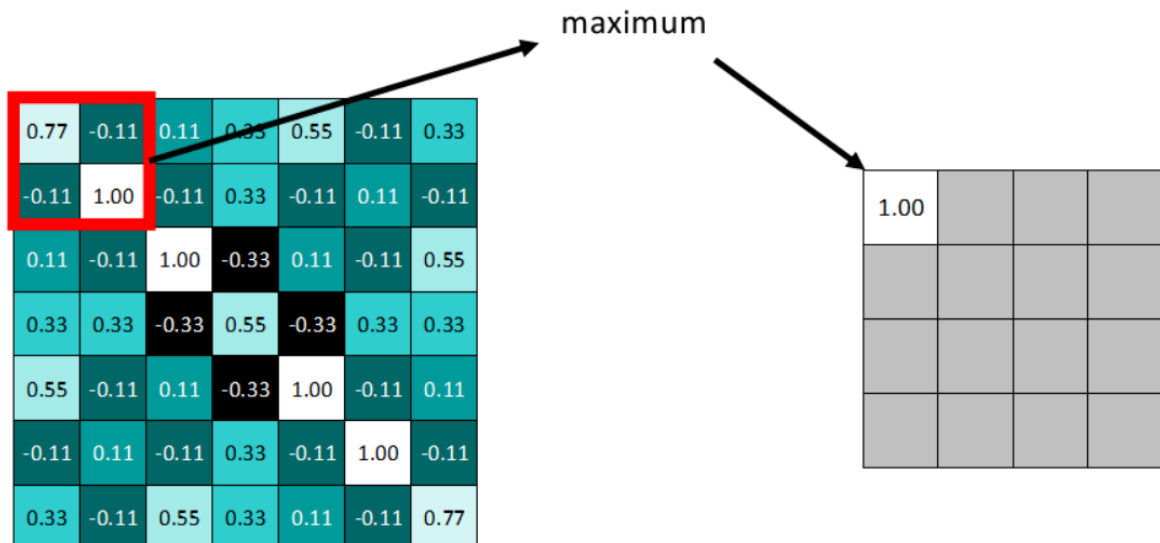
Relu函数: $f(x) = \max(0, x)$ [Relu层]

Sigmoid函数: $f(x) = \frac{1}{1+e^{-x}}$ 2



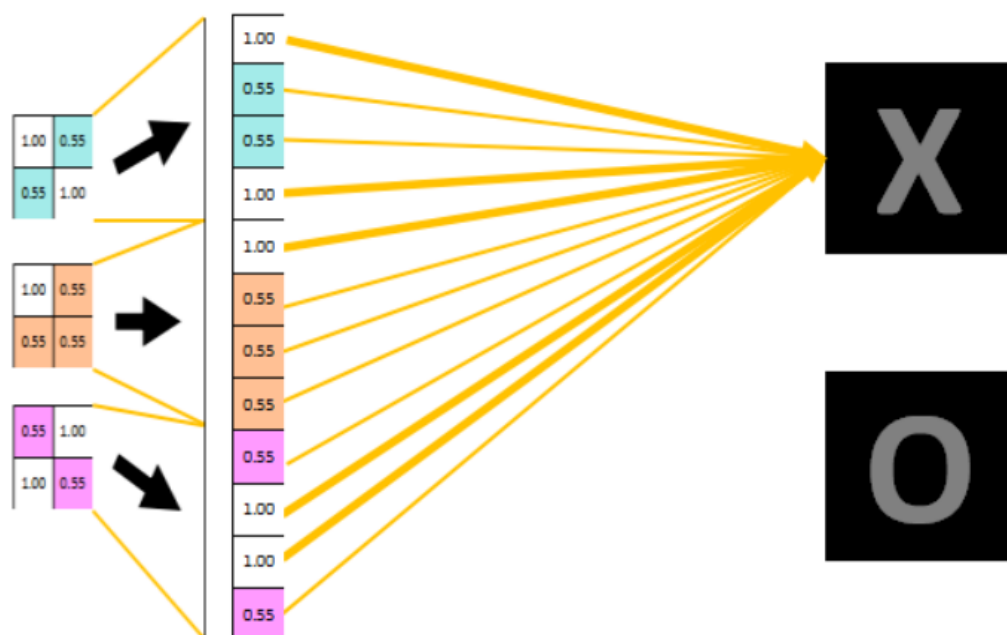
目标是减少数据量

分为Max Pooling(最大池化)、Average Pooling(平均池化)



2.3. 全连接层

目的是对特征图进行维度上的改变，来得到每个分类类别对应的概率值



展开的数据即为属于类别X的概率值，值大小也在对应X的线条粗细中表现出来了。

以上所有的操作都是对标准的原图X来进行的，因此最终分类显示即为X毋庸置疑。

2.4. 神经网络的训练与优化

训练对象：卷积核（滤波器）

训练方法：BP算法

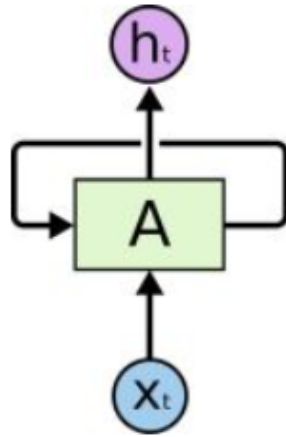
降低误差方法：梯度下降法

3. 卷积神经网络VS传统神经网络

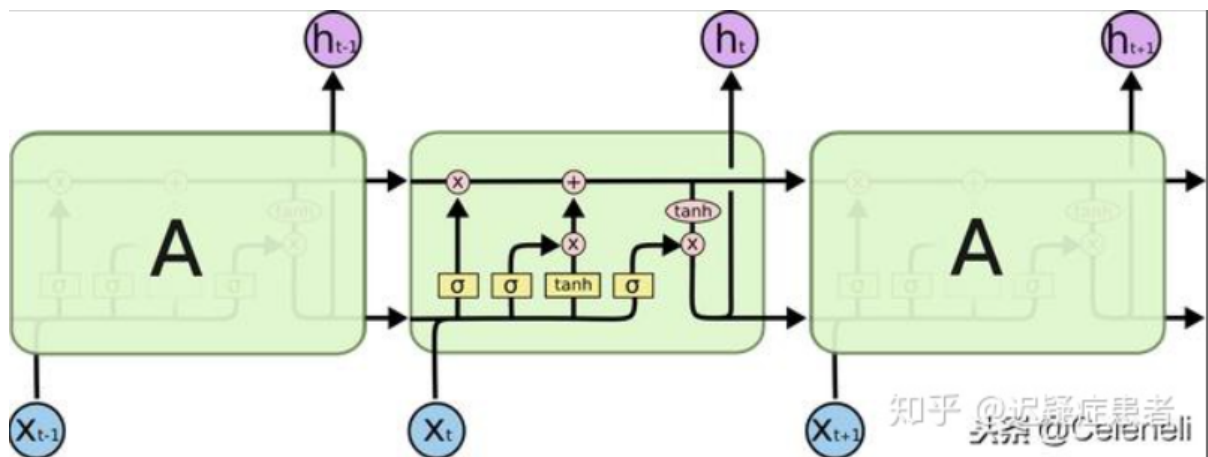
参数共享机制

连接的稀疏性

4. RNN（递归神经网络）

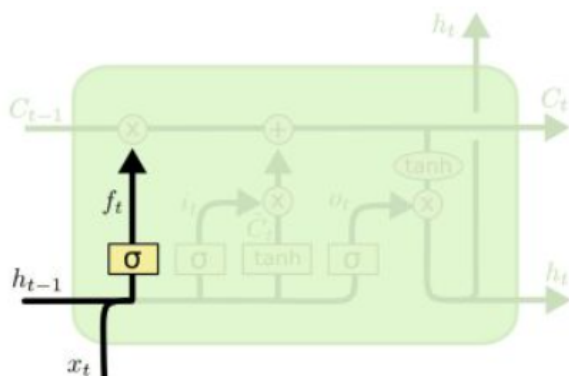


5. LSTM（长短期记忆网络）



术语：神经网络层、点态运算、向量传输、级联、复制

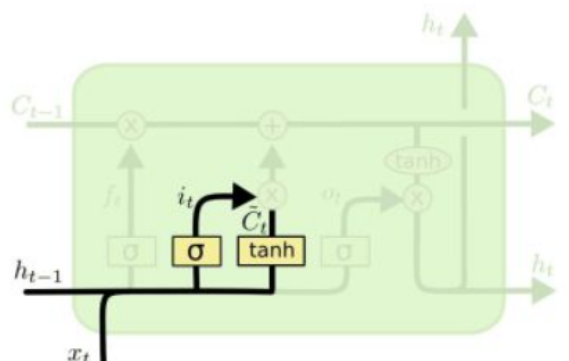
第一步是决定我们将从单元状态中丢弃哪些信息



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

知乎 @迟暮症患者
头条 @Celine

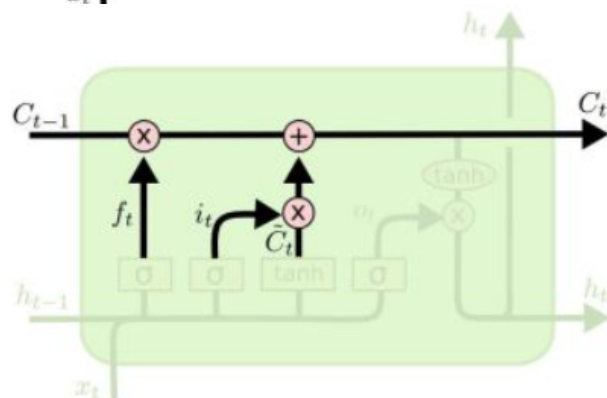
第二步是确定我们将在单元状态中存储哪些新信息



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

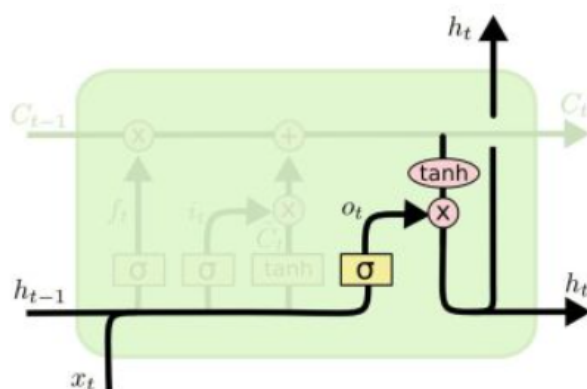
知乎 @迟暮症患者
头条 @Celine



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

知乎 @迟暮
头条 @

第三步是我们需要决定我们要输出的内容



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

知乎 @迟暮症患者
头条 @Celine

6. 自然语言处理

6.1. 词法分析

6.1.1. 分词

6.1.2. 命名实体识别

基于模型的方法

基于规则的方法

6.2. 句法分析

6.3. 语义分析

7. Typora常用快捷键

按键	效果	按键	效果
Ctrl + D	选中当前词	Ctrl + L	选中当前句/行
Ctrl + E	选中当前区块	Ctrl + F	搜索当前选中
Ctrl + B	加粗当前选中	Ctrl + H	替换当前选中
Ctrl + I	倾斜当前选中	Ctrl + U	下划当前选中
Ctrl + K	将当前选中生成链接	Ctrl + J	滚动屏幕将选中滚至顶部
Ctrl + W	关闭当前窗口	Ctrl + N	打开新窗口
Ctrl + O	打开文件	Ctrl + P	搜索文件并打开
Ctrl + 回车	表格下方插入行	Ctrl + ,	打开偏好设置
Ctrl + .	切换全角/半角标点	Ctrl + /	切换正常/源代码视图
Ctrl + Shift + -	缩小视图缩放	Ctrl + Shift + +	放大视图缩放

8. Markdown小插曲

8.1. 插入代码块

```
1 include<bits/stdc++.h>
2 using namespace std;
3 int main()
4 {
5     printf("Hello World!");
6     return 0;
7 }
```

8.2. 内联HTML代码

不会HTML，浇浇我🍷，wu'wu

8.3. 内联 $LaTeX$ 公式

$$a^2 + b^2 = c^2 \tag{1}$$

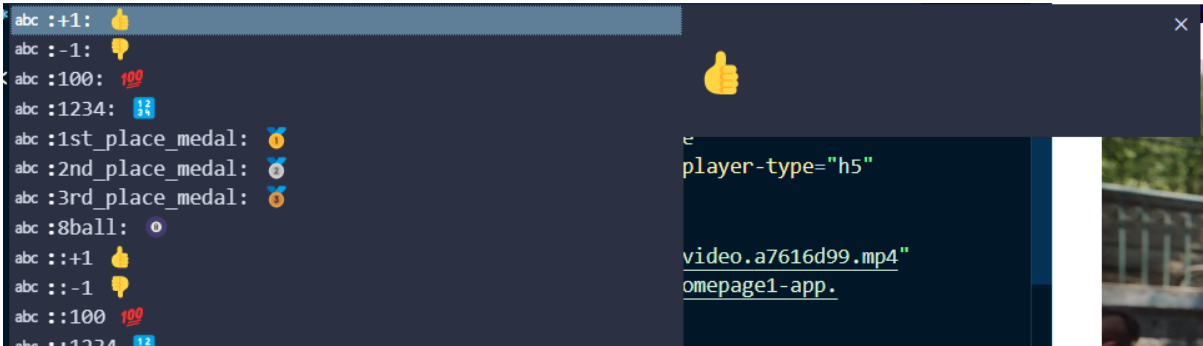
$$\mathfrak{f} \mathcal{D}[x(t)] \sqrt{\frac{3 \pi^2 - \sum_{q=0}^{\infty} (z + \hat{L})^q \exp(i q^2 \hbar x)}{(\text{Tr} \mathcal{A}) \left(\Lambda_{j_1 j_2}^{i_1 i_2} \Gamma_{i_1 i_2}^{j_1 j_2} \hookrightarrow \vec{D} \cdot \mathbf{P} \right)}} = \underbrace{\left\langle \frac{\notin \emptyset}{\varpi \alpha_{k \uparrow}} \left| \frac{\partial_{\mu} T_{\mu \nu}}{2} \right. \right\rangle}_{\text{K}_3\text{Fe}(\text{CN})_6}, \forall z \in \mathbb{R} \tag{2}$$

☑ 任务列表

单行代码 `sudo rm -rf /*`

`print('css真有趣')`

👍³ 示例如下：



markdown支持视频播放、GIF图、音频等插入

1. 指的是filter滑动 🤖 ↩
2. sigmoid函数也叫Logistic函数，用于隐层神经元输出，取值范围为(0,1)，它可以将一个实数映射到(0,1)的区间，可以用来做二分类。在特征相差比较复杂或是相差不是特别大时效果比较好 ↩
3. emoji表情，打出英文:，后面的内容高级编辑器会自动补全 ↩