

## lecture04 Backpropagation and computation graphs

创建时间： 2019/9/18 15:34

更新时间： 2020/1/13 20:58

作者： hisaishi@sina.com

---

### Derivative wrt a weight matrix

继续计算 $\partial s / \partial W$



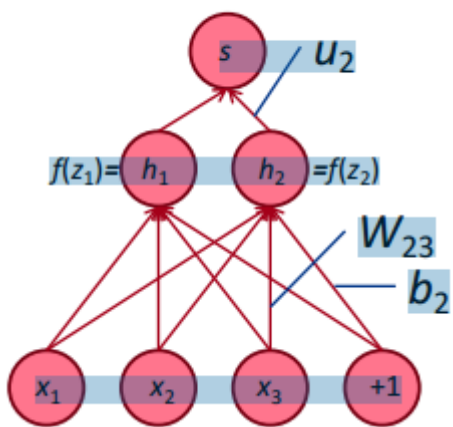
此格式暂不支持显示

先考虑单个权重 $W_{ij}$ 的导数：

$W_{ij}$ 只对  $z_i$  有贡献。例如 $W_{23}$ 只对 $z_2$ 有贡献，对 $z_1$ 没有贡献



此格式暂不支持显示



对于单个 $W_{ij}$ :



此格式暂不支持显示

对于整体权重 $W$ 的梯度，有：



此格式暂不支持显示

其中 $\delta$ 为 $(n \times 1)$ 列向量， $x$ 为 $(1 \times m)$ 行向量

### 梯度推导tips：

- 定义变量并注意他们的维度
- 求导链式法则
- 不同条件下分别考虑
- 逐个元素求偏导
- 检查维度

### Deriving gradients wrt words for window model

更新词向量的梯度可以被简单的分为每个词向量的梯度



此格式暂不支持显示

逐个更新词向量有助于命名实体的分类，但是更新非常稀疏

## A pitfall when retraining word vectors

同义的训练数据与测试数据，在情感分类中训练数据在更新过程中会不断变化，而测试数据不会

## So what should I do?

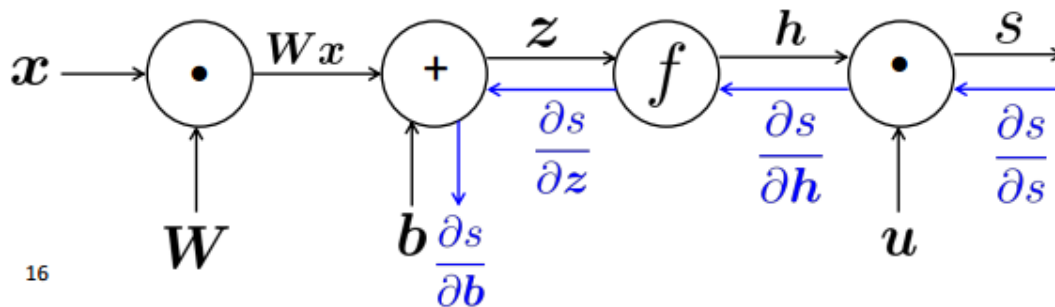
- 1.是否应该预训练词向量？。是，因为：训练算法很简单；只需在大量数据上进行
- 2.当训练有监督分类器时，是否应该更新词向量？训练数据集很小的时候不要训练词向量；数据集很大的时候最好训练-更新-调整词向量。

## Backpropagation

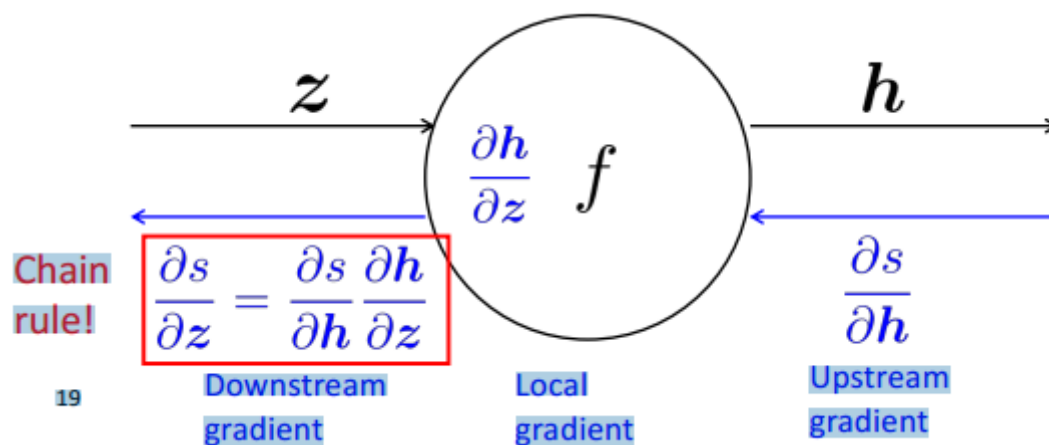
trick：在计算低层导数时，重用高层的导数，可以减少计算量。

computation graph（前向传播）：[f8e85997b9b4253bb008f05d73c1e7be.png](#)(en-resource://database/1922:0)

反向传播：沿着边反向传递梯度



以单个节点为例：同样是使用链式准则



当有多个输入时：

