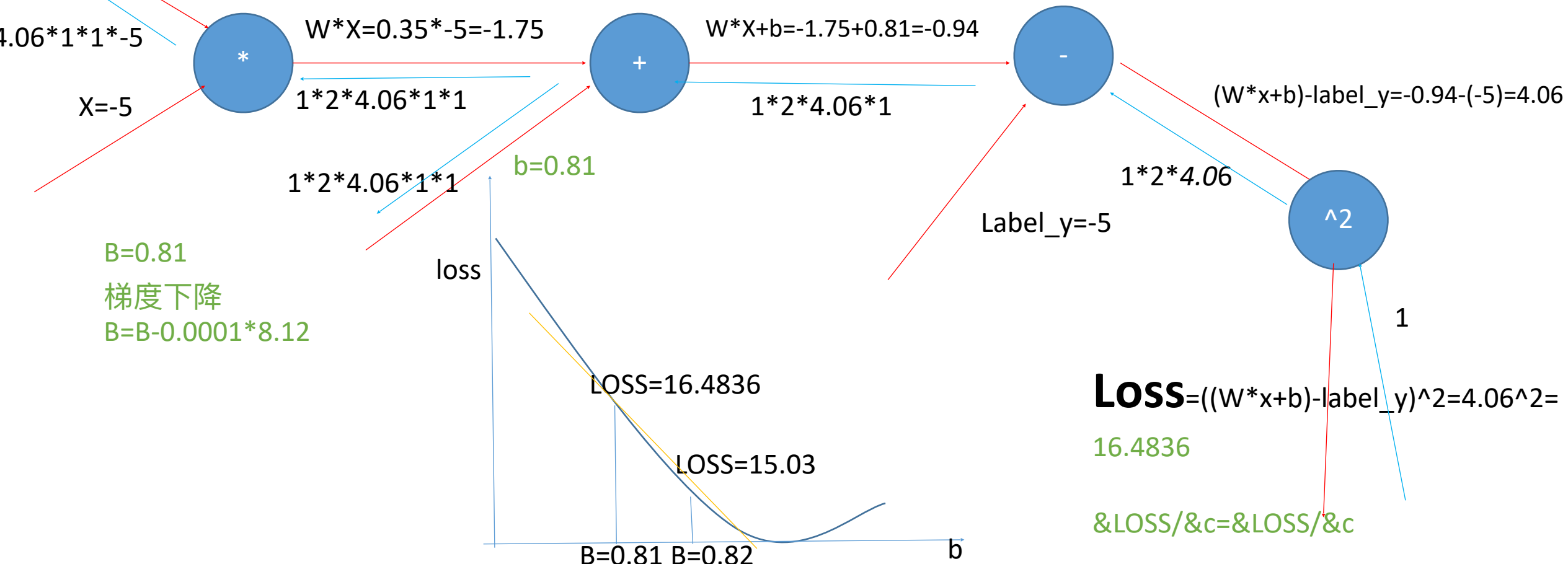


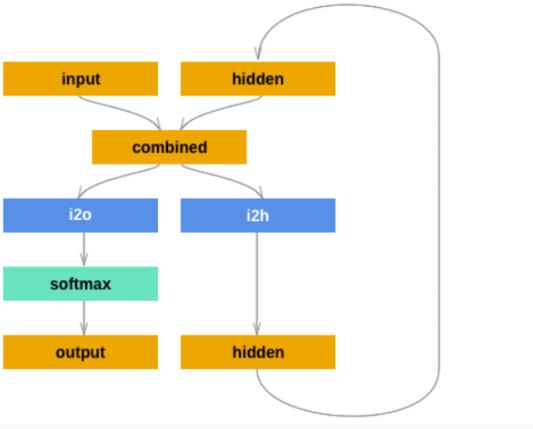
$$\text{LOSS} = (W \cdot x + b - \text{label}_y)^2 = (w \cdot x + b)^2 - 2 \cdot (w \cdot x + b) \cdot \text{label}_y + \text{label}_y^2 =$$

$$w^2 \cdot x^2 + 2wx + b^2 - 2w \cdot x \cdot \text{label\_y} - 2b \cdot \text{label\_y} + \text{label\_y}^2 = 2b - 2 \cdot \text{label\_y} = 2 \cdot 0.81 + 2 \cdot 5 = 11.62$$

$$W = 0.35 - 0.0001 * (1 * 2 * 4.06 * 1 * 1 * -5), \quad \&Loss / \&W$$



B=0.81, 我尝试调参, B=0.82, 发现loss变成15.03, 深度学习B参数: B-&Loss / &b偏导数



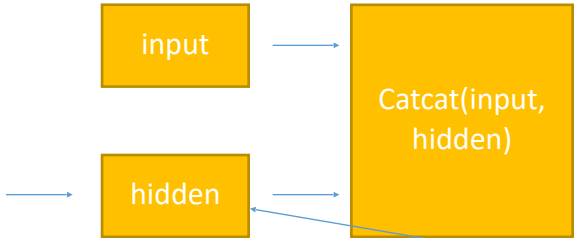
调全连接里面的W和b



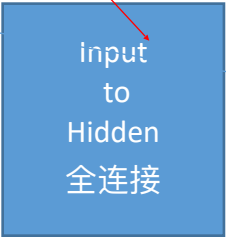
output

LOSS函数求两者差异，然后  
Backward反向传播梯度，调整模型中的权重参  
数，让Loss往更小方向变化

概率向量	真实标签
[	[
0.01,	1,
0.23,	0,
...	...
...	...
]	]

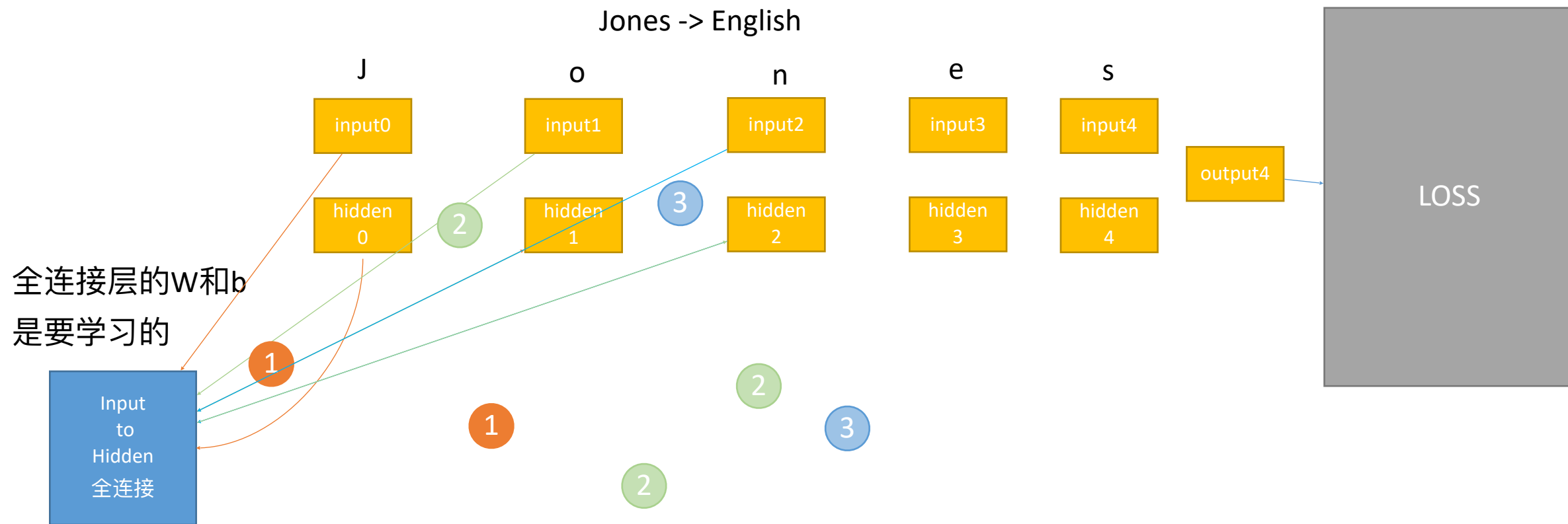
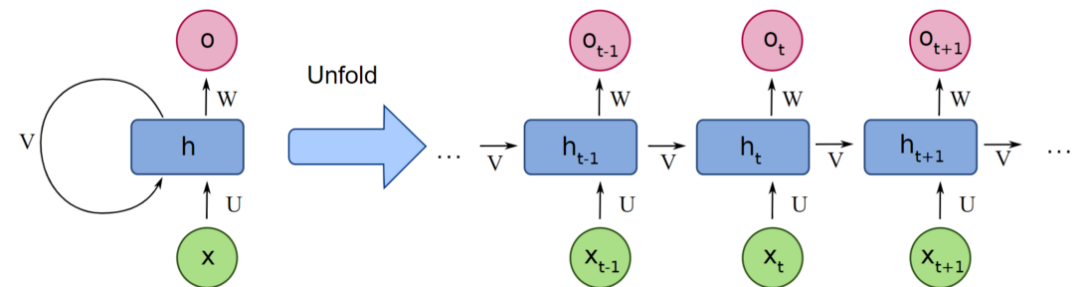


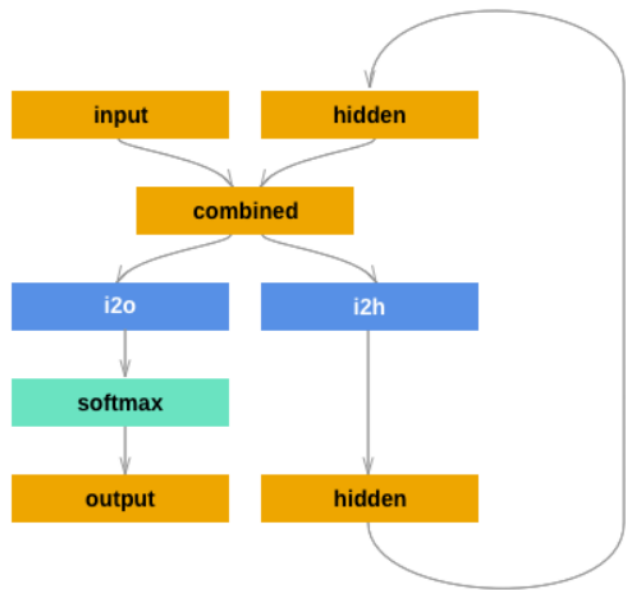
调全连接里面的W和b



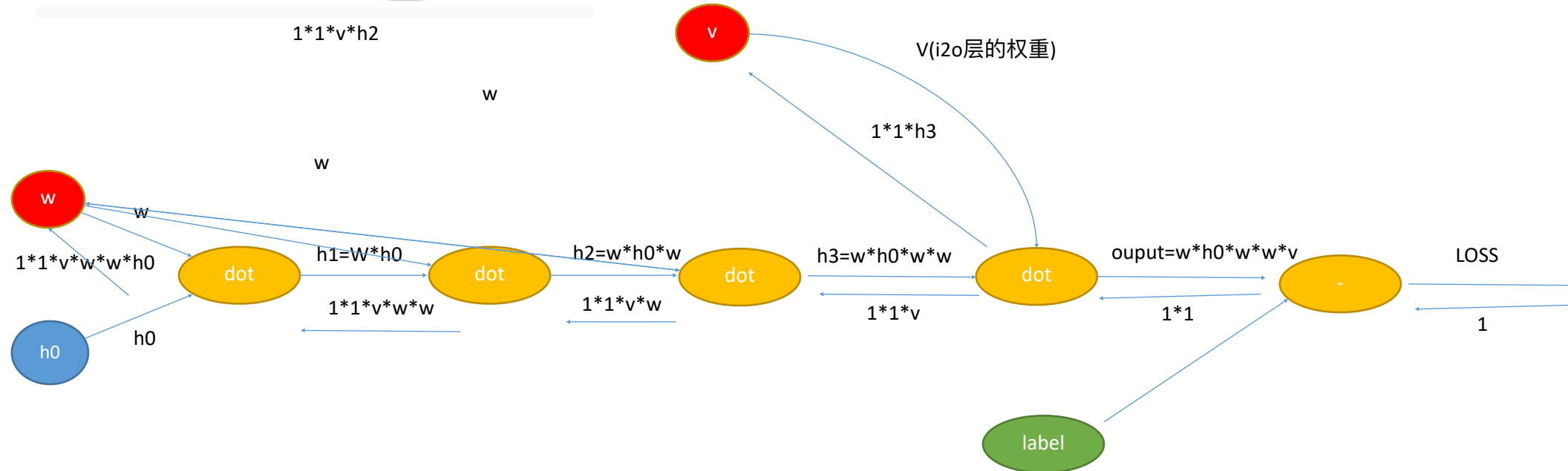
Jones -> English

- 第1次: input=(J,hidden0)
- 第2次: input=(o,hidden1)
- 第3次: Input=(n,hidden2)
- ....



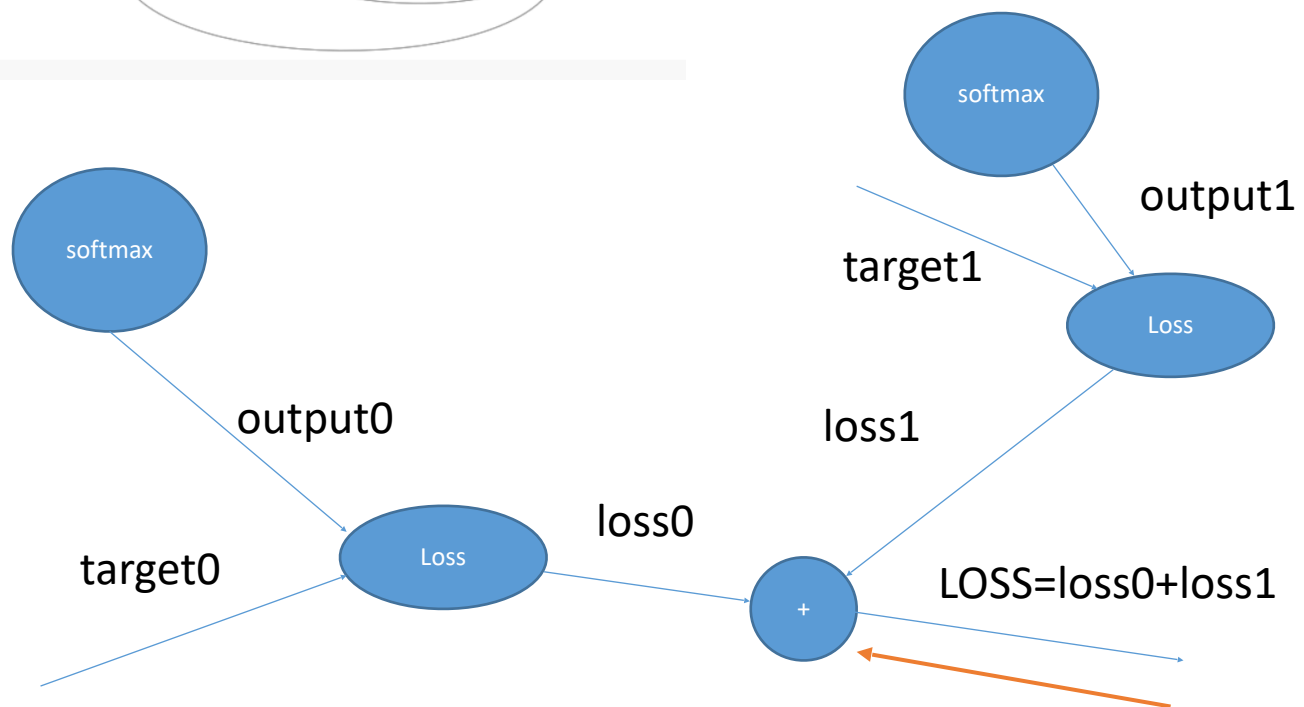


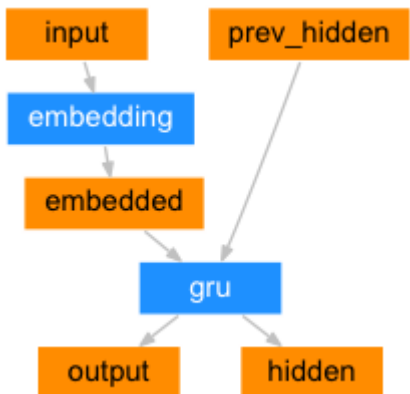
RNN缺点：梯度爆炸或者消失问题



会发现：w得到的梯度基本上是 $w^n$ 次方，那么

w如果此时 $<1$ 的，那么 $w^n$ 就可能消失为0，如果 $w>1$ 的那么w可能就爆炸成很大的数字





法语句子： Comme c'est chouette !

```
Encode_outputs=tensor([  
    [,,,..... ], #hidden_size宽的词向量 ( Comme),  
    [,,,..... ], #hidden_size宽的词向量 ( c'est ),  
    [,,,..... ], #hidden_size宽的词向量 ( chouette),  
    [,,,..... ], #hidden_size宽的词向量 ( !)  
])
```

模型就认识3个单词： Cat Dog Fish

词ID		每个词向量有几列，你自己决定			
0	Cat	0.2	0.15	0.13	0.4
1	Dog	0.1	0.01	0.32	0.15
2	Fish	0.77	0.96	0.001	0.003



注意： [0.1,0.01,0.32,0.15]  
是Parameter，是模型参数，  
是要求grad的，是要梯度下降

Embedding层

- 1， 样本输入： "Cat Dog"
- 2， 转成词ID列表： [0,1]
- 3， 循环每个词ID， 传给forward()， 第一次传forward(0)， 第二次forward(1)

以前我的理解：  
1， 把ID=1转onehot： [0,1,0]  
2， 和emb层做矩阵乘

0	Cat	0.2	0.15	0.13	0.4
1	Dog	0.1	0.01	0.32	0.15
2	Fish	0.77	0.96	0.001	0.003