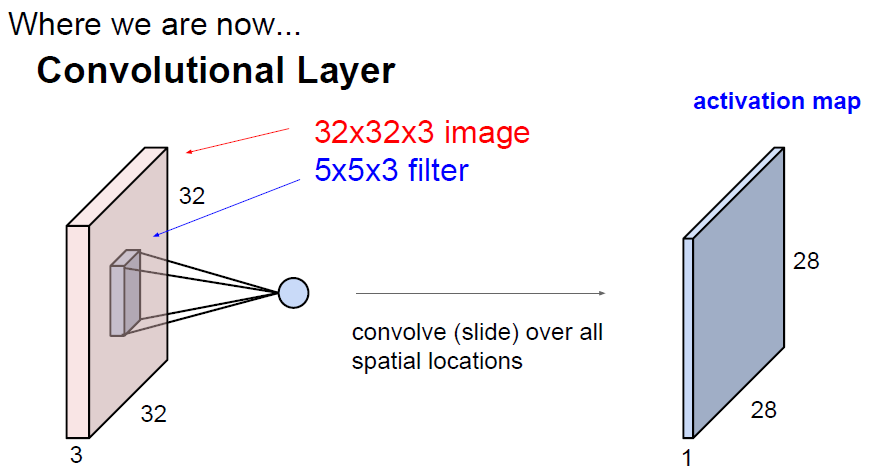
Lecture 6:

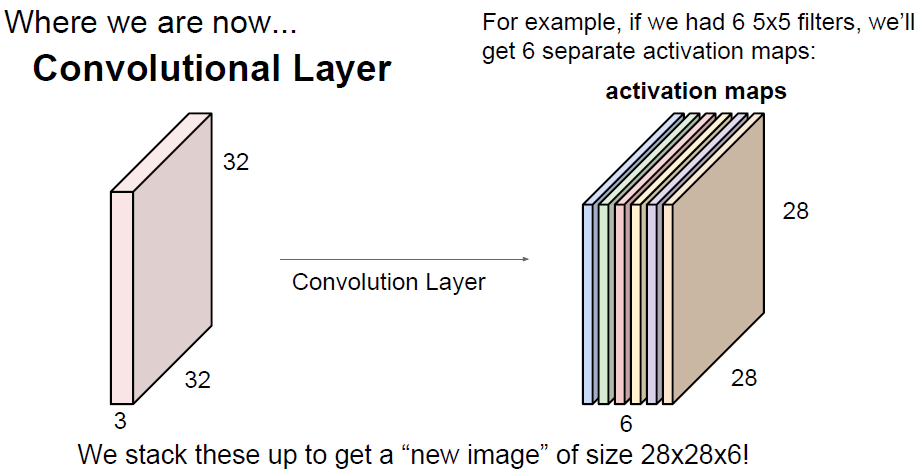
Training Neural Networks,

Part I

Notes on backprop for linear layer & vector tensor derivative linked to lecture 4 on syllabus

目前的进度：





Mini-batch SGD

Loop:

1. Smple a batch of data
2. Forward prop it through the graph(network) get loss
3. Backprop to calculate the gradients
4. Update the parameters using the gradients

下一步：

Training Neural Networks

1. **One time Setup:**

Activation functions

Pre-Processing,

Weight initial

Regularization,Gradient check

1. **Training dynamics:**

Babysitting the learning process 监控过程

Parameter updates

Hyperparameter optimization

1. **Evaluation**

Model ensembles

# Part 1 :

Activation function

Data pre-process

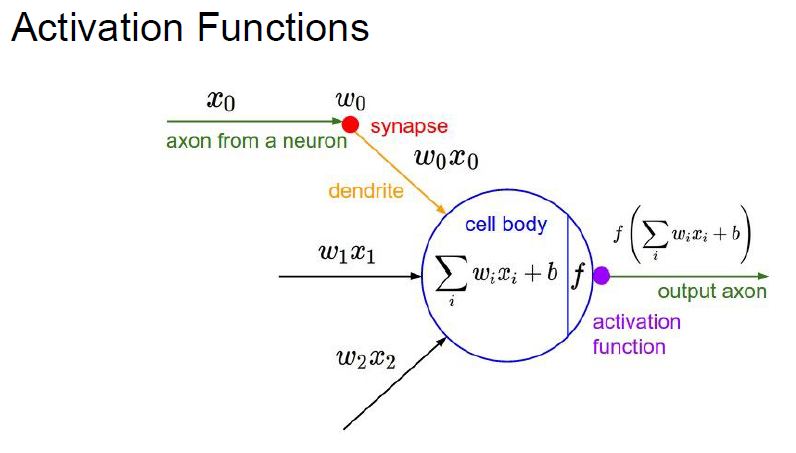
Weight initial

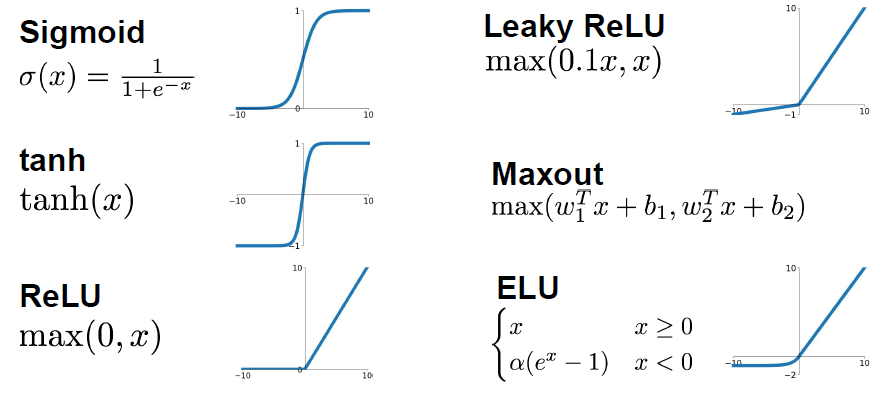
Batch normalization

Babysitting the training process

Hyper-parameter optimization

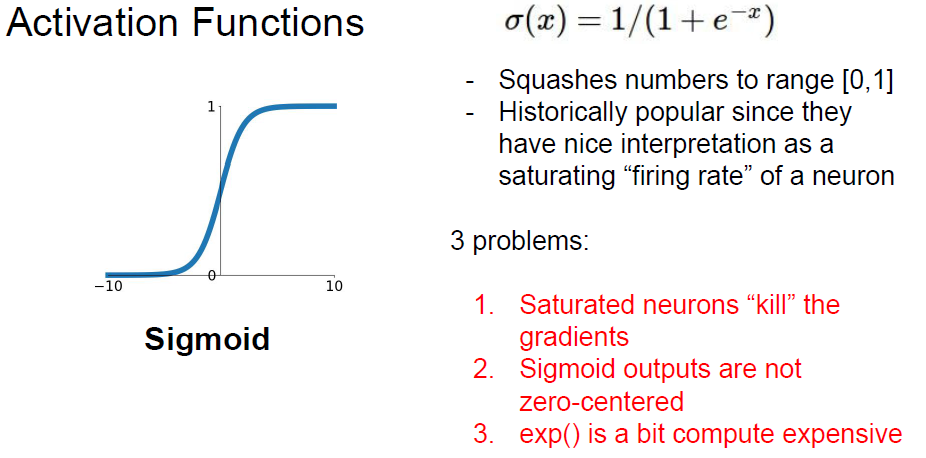
Activation Functions激活函数





## sigmoid:

把输出可以压缩到[0,1]之间，很大的值趋于1，很小的值趋于0，类似于神经元的饱和放电率

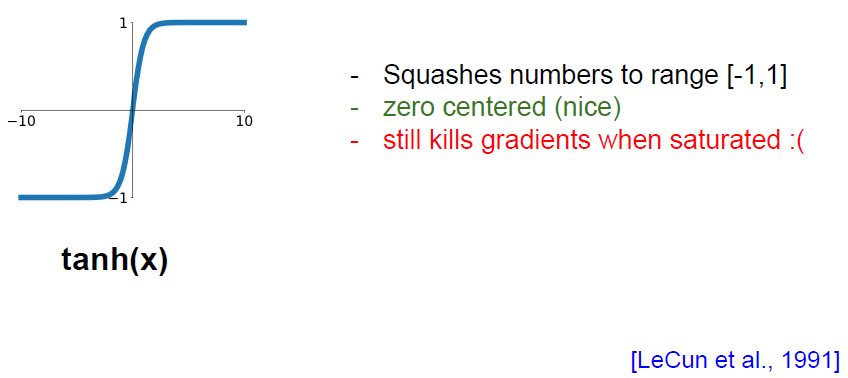




，其中,如果X都是正的，那么就是正的，而要么是全正，要么是全负，也就是说梯度更新所有权重参数w时，要么全加上正数，要么全加上负数，这样的更新效率是很低的

exp运算有一点压力，不过卷积神经网络用到矩阵的点乘，这比exp的消耗还要大…

## tanh(x)函数



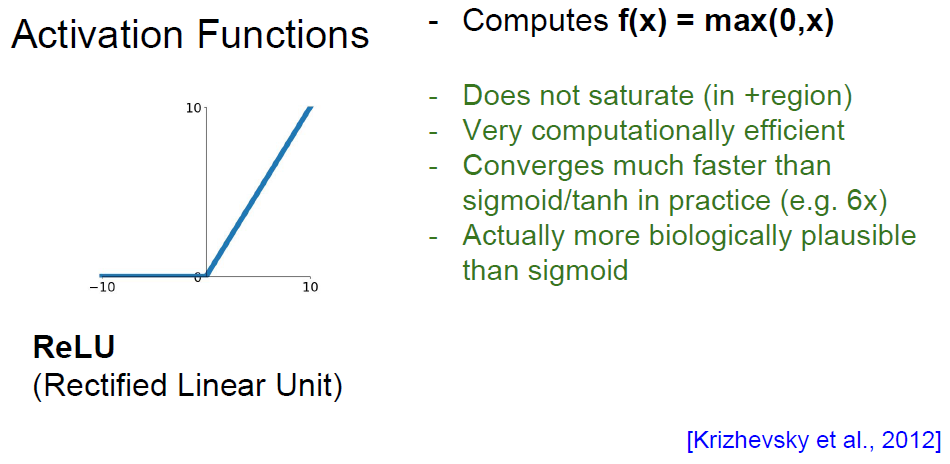
值域[-1,+1]

零中心，不像sigmoid那样如果X是正的，W求导出来也都是正的，更新慢

但仍然有梯度消失的问题

## ReLU函数(rectified linear unit)

**f(x)=max(0,x)**



不会在正区域出现饱和（saturate）现象,

计算成本也低，没有exp等运算，

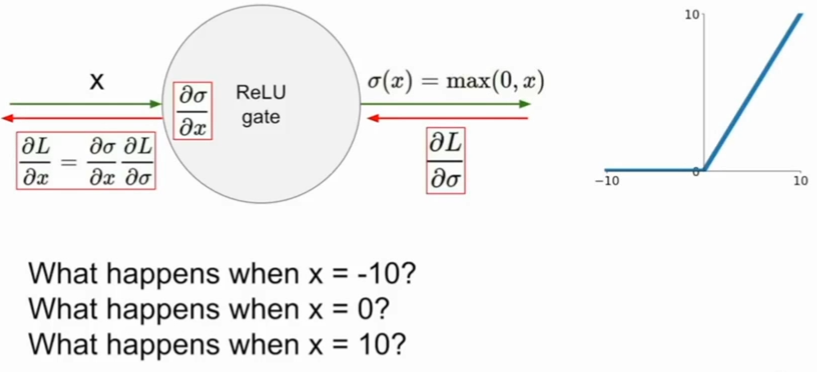
大约比sigmoid, tanh 收敛快6倍

Alexnet出现以后被大量应用

The gradient when x<0 : **0**

而ReLU也有非0是中心的问题，就是说如果X全是正的，则dw也一样是全正或全负

考虑如下例子：



X=-10, =0

X=10, =10

X=0, =0

也就是说对于小于0的X，梯度消失了，这时称为**dead ReLU**