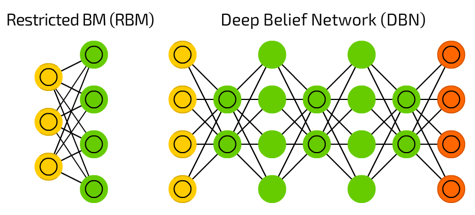
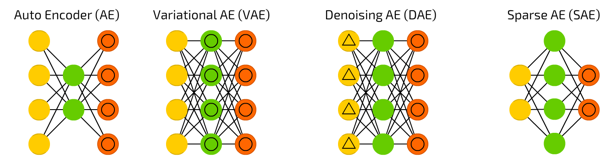
## 预测模型：

1DBN



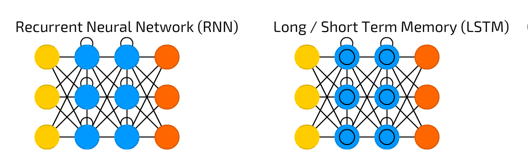
2.Stack - AE

AE



|  |  |
| --- | --- |
| 标准（数据集所具备的特征） | 结果（对应的自编码器模型） |
| 1. Sparsity（稀疏性） | Sparse AE |
| 2.Denoise（降噪） | Denoising AE |
| 3.Regularization（正则约束） | Regularized AE |

3.LSTM



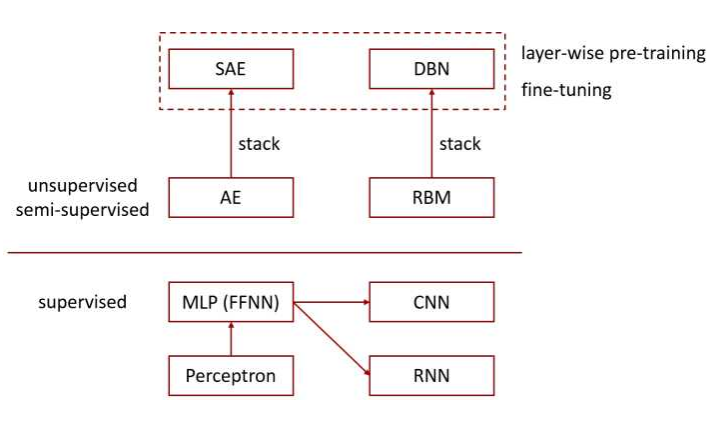
4.Ensemble Network

多个模型方法共同进行实验，通过设立不同权值，设置不同模型对于最终模型的影响。

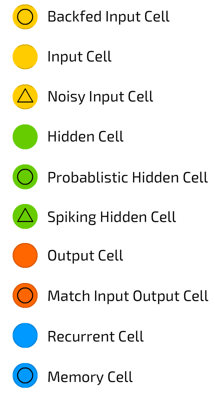
（所使用的具体方法尚未确定，与dropout优化方法有所重复）。

另，有关于使用一些优化方法比如使用Relu激活函数，使用dropout方法，优化初始化策略。在一定的条件下能够直接有效的训练深度模型。那么关于使用非监督的逐层训练方法与优化方法对于建立模型的效果可能需要进行实验比较。

引用来自Bengio的话，当所标记的数据集足够大时，使用正则化等方法所起的作用将会变小。而使用非监督预训练方法还仍将有效。

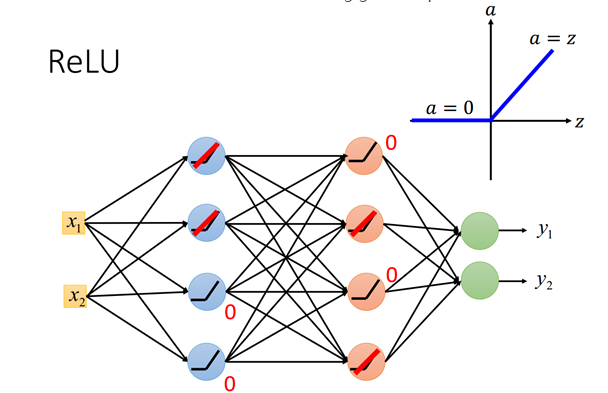
（http://yyue.blogspot.com/2015/01/a-brief-overview-of-deep-learning.html）

以监督学习和非（半）监督学习进行划分如下：

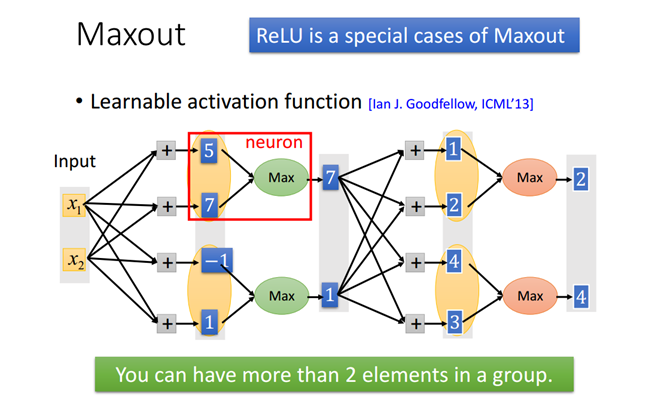
图例说明：

## 优化过程：

1.激活函数：

 1) ReLu: f := max(0,x).

2)Maxout： z\_i = Εx \* w\_i. f := max(z\_i).

2.梯度

1) gradient decrease

2) stochastic gradient decrease

3)mini-batch gradient decrease.

每一次迭代计算mini-batch的梯度，然后对参数进行更新



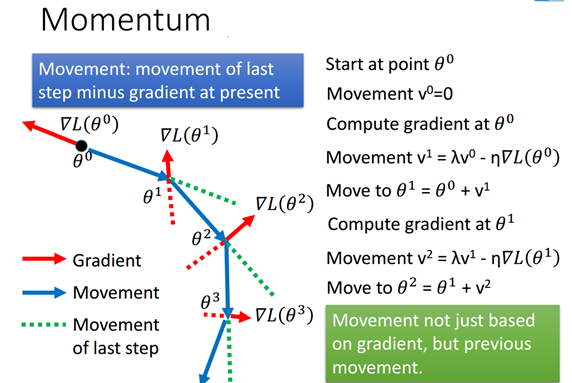


其中，μ是学习率，g\_t是梯度 SGD完全依赖于当前batch的梯度，所以μ可理解为允许当前batch的梯度多大程度影响参数更新

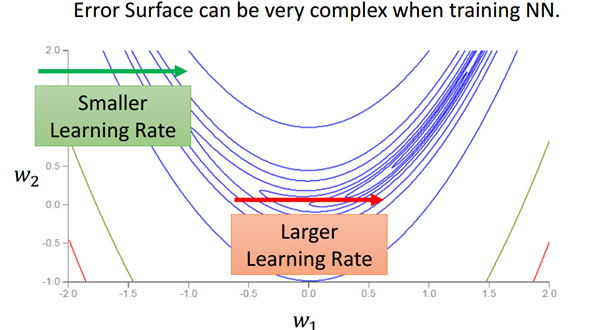
4）Momentum

参考真实物理世界中的这个惯性，在深度神经网络训练中引入这个惯性的概念尝试避免陷入局部的最小值，导致训练中断。（不保证一定不会陷入局部最优）。

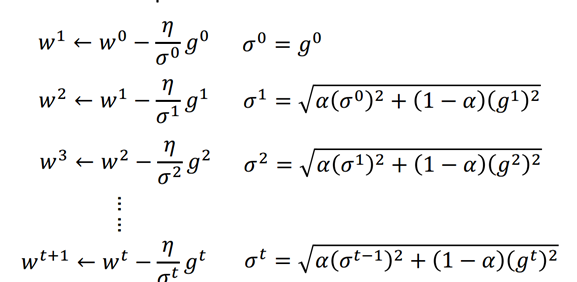
momentum每次移动需要考虑两个方面，第一个是梯度的方向，第二个就是所谓的惯性。



3.调整学习率进行优化

 1）Adagrad

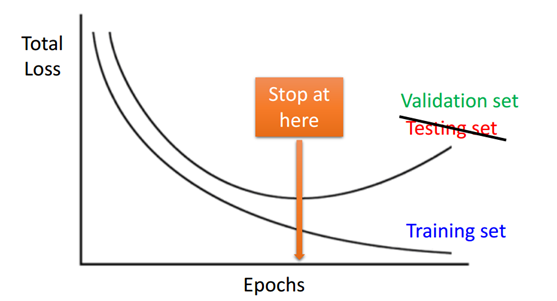
2）RMSprop：于不同的梯度下采用不同的学习率。



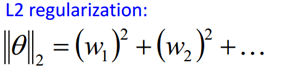
Adam : RMSProp + Momentum.

4. 快速结束（early stoping）

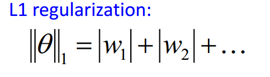
为了让测试集上的正确率和训练集上的正确率不要相差太多，Early stoping是其中的一个方法。

由于训练集具有一定独特的特性，所以可能会与测试集有一定差异。这会造成随着训练次数的增加。虽然训练集的Loss 逐渐减少至收敛。但是测试集会出现反而增加的情况。所以在合适的地方停止训练将会提高模型精度（在每一次训练结束后都计算验证集的错误率，在错误率上升时停止训练）。

5.正则化。

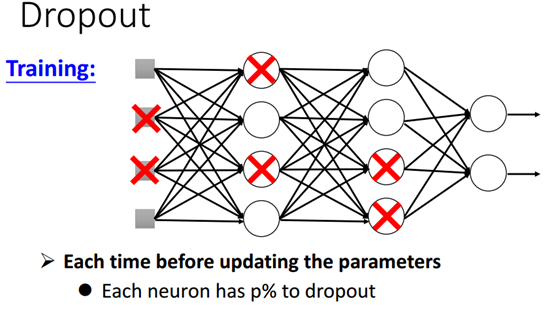
L2 正则化：

L1 正则化：



6.Dropout

每一次的参数更新过程后，会将隐层节点以概率p失活不进行下一次的模型训练。



于测试实验时将会在每一个节点乘上1-p %生成预测模型。

模型编码：