# Deep Learning for Vision Tricks of the Trade

## CHOOSING THE ARCHITECTURE选择合适的体系结构

Task dependent任务依赖

Cross-validation交叉验证

[Convolution → LCN → pooling]\* + fully connected layer

Local contrast normalization（局部对比归一化）

The more data: the more layers and the more kernels更多数据，更多层，更多卷积核

Look at the number of parameters at each layer查看每一层的参数数量

Look at the number of flops at each layer查看每一层的失败数

Computational cost 计算成本

Be creative要有创意

## HOW TO OPTIMIZE 如何优化

SGD (with momentum) usually works very well

Pick learning rate by running on a subset of the data

通过在数据的子集上运行来选择学习速率

Bottou “Stochastic Gradient Tricks” Neural Networks 2012

“随机梯度技巧”2012年的“随机梯度法”

Start with large learning rate and divide by 2 until loss does not diverge

从大的学习速率开始，除以2直到损失没有发散

Decay learning rate by a factor of ~100 or more by the end of training

在训练结束时，衰减学习率是100或更多

Use non-linearity使用非线性

Initialize parameters so that each feature across layers has similar variance. Avoid units in saturation.

初始化参数使得层上的每个特征都有相似的方差，避免神经元单元饱和。

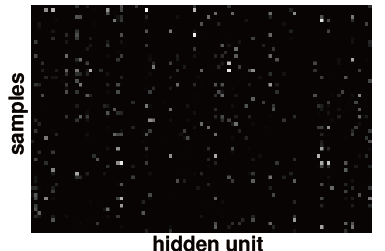
## OTHER THINGS GOOD TO KNOW

Check gradients numerically by finite differences

通过数值有限差异检查梯度

Visualize features (feature maps need to be uncorrelated) and have high variance.

可视化特征图（特征图谱需要不相关并且具有高偏差）

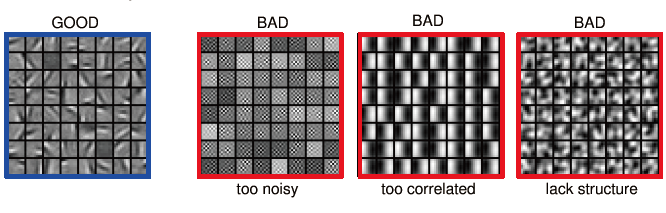


好的状况：通过将隐藏层的神经元单元输出的数值使用图像化表示成为图片的形式分析数据。如果神经元**输出的数值跨神经元，跨样本稀疏表示。**



坏的状况：许多隐藏的单位忽略输入或表现出强烈的相关性

可视化参数



好的参数可视化后展现的结构并不相关

Measure error on both training and validation set.

在训练和验证集上测量错误

Test on a small subset of the data and check the error → 0.

测试一小部分数据并检查错误。

## WHAT IF IT DOES NOT WORK?

### Training diverges（训练发散）

Learning rate may be too large → decrease learning rate学习率太大—>减小学习率

BPROP is buggy → numerical gradient checking（数值梯度检查）

### Parameters collapse/loss is minimized but accuracy is low

参数loss值被最小化，但是准确率很低

Check loss function:检查loss函数

Is it appropriate for the task you want to solve?这对于你需要解决的问题是否合适

Does it have degenerate solutions?是否有别的解决方案

### Network is underperforming（网络表现不佳）

Compute flops and nr. params. → if too small, make net larger（增大网络）

Visualize hidden units/params → fix optmization（可视化隐藏层单元/参数）修复优化函数

### Network is too slow（网络迭代速度太慢）