Neural Network

1. 文本

   中度可信度描述已自动生成**McCulloch&Pitts 感知机**

**更新权重**：（w = w-kdw） **更新阈值**：θ( τ +1 ) = θ( τ ) – ηt (t = t - kdt)

感知机的局限：只可以分线性可分的问题

反向传播算法：sigmoid导数 = s(x)(1-s(x))

白板上写着字

描述已自动生成

**神经网络输入预处理：**Encoder原始数据，（对于无大小关系的类别变量用onehot编码，对于有大小关系（离散变量）的类别变量用label encoding硬编码，离散变量，连续变量，循环变量用sin，cos或Overlapping soft intervals）

**神经网络设计：**1.适合的大小，否则回过拟合 2.训练集小，尝试数据增强或加噪声 3.调整反向传播训练参数：用safe initial value，avoid momentum（避免动量）, use pattern mode (stochastic随机梯度下降) updating of weights 4.输出编码 5.关注决定边界（decision boundary）的weighting pattern

**反向传播的局限**：1.训练慢 2.架构不容易预先确定，隐藏层不好确定 3.Moving target 4.Herd effect 5.Step size

**隐藏神经元数量 1.**successive layers of processing neurons should have fewer neurons（连续处理层有较少神经元）2.第一层大于等于第二层 3.最后一层有最多特征数量的神经元

**欠拟合怎么办：**1.加一个额外隐藏神经元 2.加10%额外隐藏神经元 3.隐藏神经元翻倍

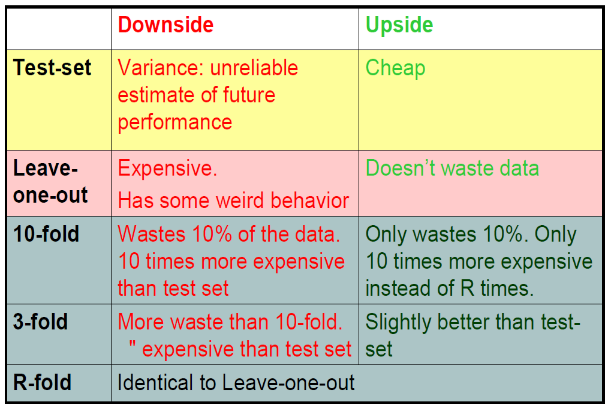
**Cascor correlation NNs**

1. hidden units are added only one at a time and do not change after they have been added(添加神经元)
2. learning algorithm—creates and installs the new hidden units. For each new hidden unit, the algorithm tries to maximize the magnitude of the correlation between the new unit's output and the residual error signal of the network.（对新隐藏层最大化correlation）

优点：1.比普通反向传播快10倍+ 2.由网络自己决定网络大小和拓扑topologies结构 3.适用于新增数据的网路训练

缺点：1. Old units weight frozen. 就神经元可能效果不好，导致网络变大 2.使用correlation measure. Forces the hidden units to saturate, which produces jagged edges in the network outputs.

**Resilient Backpropagation**有弹性的反向传播

只使用梯度的方向，因为梯度大小可能差且有噪声。

1.两次方向相同增加step 2.两次方向相反减小step 3.Weight 总是改变

Rprop优点：不需要设置step size

**Casper** 和cascor区别：1.Weight decay 2.Adaptive learning rate 3.新神经元加入后重置learning rate

**Layered casper** 和普通casper区别**：**设定每层神经元数量，当该层神经元数量到达时，再增加深度

## Validation and cross validation

Validation强调overfitting。Internal Validation: 统一数据集cross-validation。External Validation**:** 用别人的数据调

cross-validation：当证明模型没有过拟合时使用

**Leave-one-out cross validation**

有多少数据集就分成多少类

**cross validation优点**

1.避免过拟合2.比较不同的学习算法3.选择隐藏神经元的数量4.特征选择5.选择多项式的degree

# *Deep learning*CNN

特点：稀疏相互作用sparse interactions，权值共享

1.Filer的数量等于你想检测的特征的数量 2.同一feature map下的神经元权值共享 3.Feature map的数量等于想要提取出特征的数量 **Batch normalization**减少internal covariate shift（内协变量移位）

**RNN** 1.使网络有记忆2.输入输出有时序性的3. 神经元有反馈链接（hidden state）给输入，Hidden state代表了网络目前的memory和当前输入的状态h t = w hh h t 1 + w xh x t + 𝐛𝐡. Whh: hidden to the hidden weight, wxh: input to the hidden weight. Bh bias.

**局限：1.**单个state和weight向量用于学习语句关系时，总是被更新导致模型对于长文本效果差 2.RNN不稳定，反向传播会导致梯度消失和梯度爆炸

**让RNN稳定的方法**1.Gradient clipping：缩放（rescaling）梯度使他的norm在特定值来防止梯度爆炸 2.Regularization：在weight中加penalty和weight decay原理一样 3.Input reversal：正反双向学习输入和目标词汇——更短的词间距 4.Long short term memory RNN（LSTM）

**LSTM ——将RNN中nonlinear unit换成memory block**

1.Input门控制哪些新输入进入当前隐藏层——基于之前memory cell的内容，hidden unit 和当前输入2.Output门控制哪些信息从memory cell中进入current hidden unit3.Forget门控制什么时候memory cell遗忘4.Input modulation门调制允许更快下降的信息进入memory cell**双LSTM解决对长文本遗忘最开始信息的问题**

## 无监督学习 无监督预训练：有限的labelled数据和大量与labelled data相似的unlabelled data.1.学到输入的分布有助于以后学习2.DNN参数初始化很重要，帮助避免overfitting

**非监督贪心逐层训练Greedy layerwise unsupervised pretraining在unlabelled data预训练**每次只训练一层，固定参数后再训练下一层

## 生成模型generative model

**VARIATIONAL AUTOENCODERS VAE** Encoder：将输入编码为编码code Decoder：将编码code解码为目标数据，比如照片

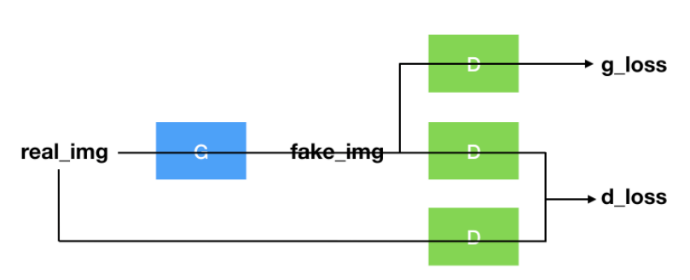
VAE与普通autoencoder区别：让编码符合某种分布如正态分布，目的是之后只需要输入符合正态分布的数据即可获得解码的目标数据。

**GENERATIVE ADVERSARIAL NETWORKS GAN**

计算loss：MSE between pixel values

Generative Model：通过随机噪声生成模型，使用反卷积网络Deconvolutional/Conv Transpose NN

Discriminative model：辨别生成的照片，使用CNN

**1）GAN训练失败的标志1.**D loss or G loss变0 2.D或G loss变得最大并且不变 3.梯度的范数/模norm大于100 4.G loss不断减小时，他在用垃圾愚弄D D loss有低方差且逐渐变小 或 方差很大且不断变化 是正常现象

**2）G loss：max(log(D)), min(log(I-D))会梯度消失**

**3）正则化输入**将输入照片正则化在[-I，I] Tanh 是G输出的最后一层

**4）G的输入**：使用正态分布而不是均匀分布

**5）Batchnorm批归一化**

使用minibatch梯度下降，一个batch中要么是真图片要么是假图片

当batchnorm不可用时使用实力规范化instance normalization——减平均数除标准差

**6）避免稀疏梯度Avoid Sparse Gradients**

使用leakyrelu，对于下采样使用average pooling，cnn+stride，对上采样使用cnnTrans+stride

**7）Use Soft and Noisy Labels（使用光滑和带噪声的标签）**

对标签进行平滑 比如真实标签为1转为0.7-1.2 在训练D鉴别器的时候，偶尔翻转label，即fake->real, real->fake注：很难知道模型什么时候训练好，不时要看sample，不能只看loss