卖点：
- 可以建模 $p(\tilde{y}|x,y)$ 的噪声
- 加一层 softmax 就可以.

以前算法-利用EM算法.

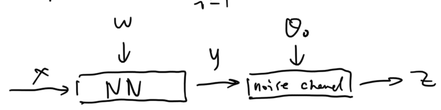
Intro:

以前的算法利用 $\tilde{y}$ 和 y 关系为 asymmetric 将 真实标签为隐变量.

$$p(z=j|x;w,\theta)=\sum_{i=1}^{K} p(z=j|y=i;\theta)\,p(y=i|x;w)$$

↑ 参数不好优化，w不好求出.

假设



用EM算法求解.即

$$\arg\max_{w} \; p(y_t=i|x_t,z_t,w_0,\theta_0)\cdot p(y|x;w) \Rightarrow$$

- θ 有显式表达 $\theta(i,j)=\frac{\sum_t C_{ti}\mathbb{1}_{z_t=j}}{\sum_t C_{ti}}$.
- w通过梯度下降. $\frac{\partial S(w)}{\partial u_i}$.

(变种) ① 不学 θ_0，设置一个验证集手工标.

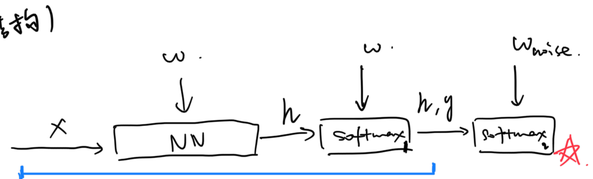
② 蒸馏学习，先用大网络带噪学习. 后将前面网络参数 w_0 用于初始化小网络.

(缺点)：EM算法易陷入局部最优. EM易收敛. asymmetric 标签假设太强.

提出算法 — 只基于DNN+梯度下降

Method:

(结构)



S-model：假设 $\hat{y}$ 是 asymetric的. 所以. w_{noise} 是不含参数 u_{ij} 的.

C-model：假设 $\hat{y}$ 是 instance-dependent的. w_{noise} 含参数 u_{ij}.

$$\text{softmax}=\frac{\exp(u_i^T h+b_i)}{\sum_i \exp(u_i^T h+b_i)}$$

> 直接最大似然梯度下降

(初始化)

w_0 为 1 stage 的模型参数　　w_{noise}：u_{ij} 置 0.；b_{ij}：1 stage 混淆矩阵

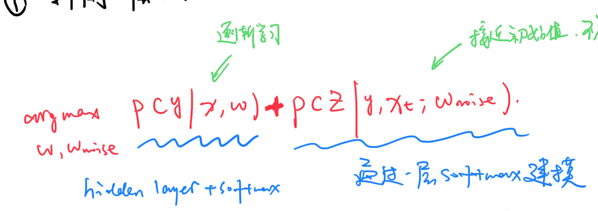
(步骤)

① 首先带噪训练 DNN，后假设 $\hat{y}$ 为 y，初始化出下，即 softmax_2 的偏置参数

② DNN + softmax_2 再训练一次. 如果 S-model 就使 u_{ij} 恒等 0.

(优点)

① 利用一层 softmax 显式建模 $p(\tilde{y}|y,x)$，并且初始化使得模型可能学到 $p(y|x)$.



假设第二项变化小，那么 $p(y|x,w)$ 和 $p(\tilde{y}|x,w,w_{noise})$ 是 statistically consistent 的

② 可扩展性好，仅需要做：① 统计 T ② 加一层 softmax.

实验

1. 比较对象.

baseline：带噪训练的 DNN.

reed-soft：$\beta=0.95$.

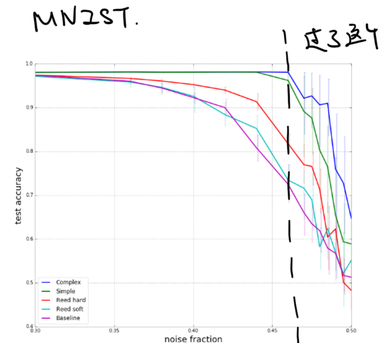
reed-hard：$\beta=0.8$

S-model

C-model.

2.

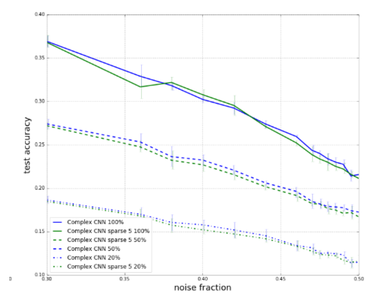
MNIST.



(c) 100% dataset

过了这个点准确度下降？值得研究.

CIFAR-100.



在类别数量大 (100) 的情况下.

下只记录每一类前 k 名最易转移的标签概率.

↓

和 dense 初始化相比没有较大的性能下降.