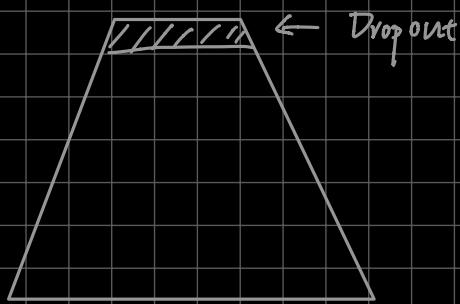


Dropout: 加在 网络的 顶部 而非 底部

① 底部应尽可能保存输入中信息的丰富性



② 顶部加入 dropout 相当于引入随机的结构，使一个 model 等同于多个相近但 task-specific 的特征 且有 不同的 network 叠加后
由顶层决定 网络结构不同

取均值，减少了 model 对某些特征学习不充分的现象。

Dropout rate: 0.1 - 0.5 间，0.5 常用

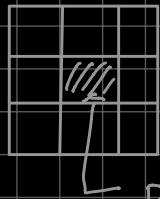


调整：明显过拟合现象减轻 \Rightarrow 降低 dropout rate

过拟合现象仍严重 \Rightarrow 增大 dropout rate

△ 卷积层一般不使用 dropout，Dropout 应用于全连接层
 \Rightarrow Dropout 用来防止 参数过多而产生的过拟合。而卷积层
模型过大。

的优点就是能提取输入的空间层次的信息，且
具有平移不变性。因此相同的像素共享很多
信息，即使



中任何一个被删除，它们

所包含的信息可能仍然会从相同的像素 向下一层

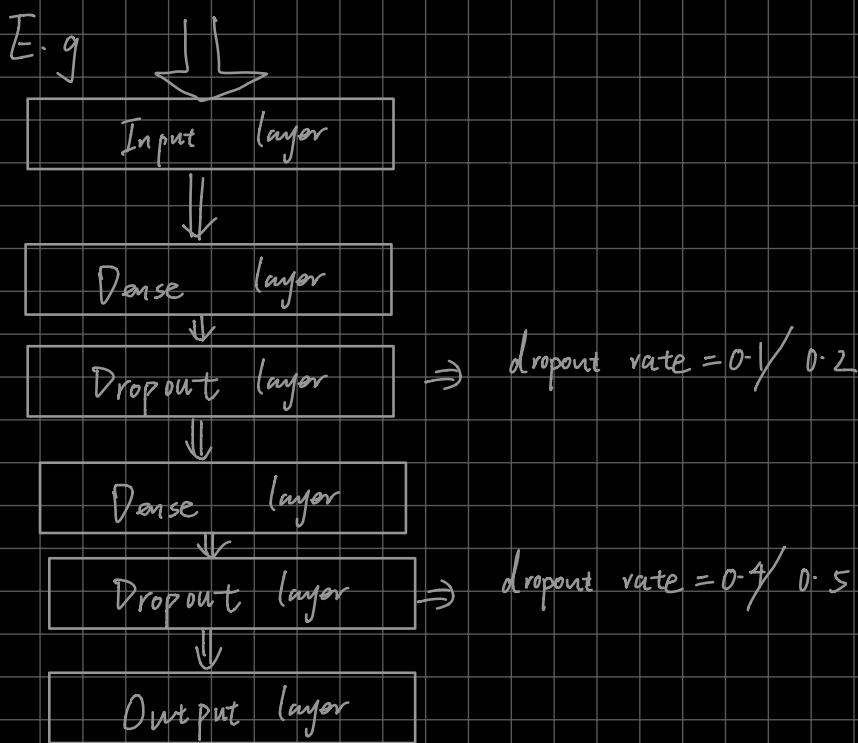
传递。因此此时 Dropout 作用相当于加强模型
对 噪声的鲁棒性，而非如全连接层一样
的效果。



即使当前数据规模远大于参数规模 / 不会出现
 过拟合效应，也应该^{0.01 等于 drop out rate}加入 Dropout (或一些其它形式
 的小噪音) 来 “平滑” 代价格局，可以 ① 可快训练过程
 ② 有助于处理异常值，并防止网络中出现极端权重结构勾
 打破数据中的偶然相关性

Spatial Dropout: 用于 CNN 的 Dropout. \Rightarrow

Spatial Dropout 会随机丢弃 整个特征图 (而非
 单个元素)



Recurrent Dropout = 用于 RNN. LSTM. GRU 的 dropout.

对不同的时间步使用相同的 循环层 dropout 掩码，
 使网络朝着正确的方向学习。

Batch Normalization: 加在网络的底部

作用: ① 防止 ICS, 提升模型训练速度

Internal Covariate Shift (内部协变量移位)

② 防止网络训练进入梯度饱和区, 减缓
网络收敛速度.

method:
① Batch Normalization
② 使用 Relu.
Leaky Relu 代替
tanh. sigmoid

③ 降低模型对参数敏感度 (模糊共享效应),
简化调整过程, 使网络学习更加稳定.

参数
走参数

method

① 权重初始化方法 ② 学率 ③ Batch
(e.g: Xavier) Normalization

ICS: 第 L 层的参数变化通过: $Z^L = W^L \cdot X^L + b^L$

$A^L = g^L(Z^L)$, $X^{L+1} = A^L$ 影响 L+1 层输出

数据分布, 而 L+1 层需要不停地去适应这样的变化

此即 Internal Covariate Shift.

原理: 简约白化(Whitening) + 线性变换

⇒ 每个特征均值为 0, 方差为 1

$[u_i = 0, \sigma_i = 1 \text{ (for all } i\text{)}]$

使操作后数
据恢复表达
能力。

PCA 白化：所有特征分布
均值为 0，方差为 1。
 \uparrow

Whitening: $\xrightarrow{\text{PCA}}$ PCA whitening: $\mu = 0, \Gamma = I$

$\xrightarrow{\text{ZCA}}$ ZCA whitening: $\mu = 0, \Gamma_i = \Gamma_j = \Gamma_k \dots$

ZCA 白化：所有特征分布
均值为 0，方差相等

\triangle ZCA 白化 = PCA 白化 + 旋转操作 \Rightarrow 使得处理过数据更接近
原始数据。

作用：
① 使输入特征分布具有相同均值、方差
降低特征之间相关性 / 未离合性

② PCA 白化还可起降维作用

缺点：
① 计算成本高

② 降低了网络表达能力 \Rightarrow 底层多数据

信息被白化操作丢失掉

防止过拟合 && 提高网络准确性

① 最佳方法：向网络输入大量且不重复训练数据

② 设置合适 batch size。

batch size 过大使得梯度下降速度减慢
易对网络准确性产生负面影响。

batch size 过小不利于充分利用 GPU 并行性

③ 设置合理的学习率 (Learning Rate)

过大 \Rightarrow Divergence / 网络不收敛

当学习率一改变后网络的行为不可预测

过小 \Rightarrow 训练缓慢

如何选取：关闭梯度裁剪后，找出学习率
这样能看出误差爆表

最大值 $\alpha_{\max} \Rightarrow$ 训练过程中不会让误差爆表的上限值

$\alpha \leftarrow \alpha_{\max} \cdot 0.9 (0.95)$ ，很可能接近最佳学习率。

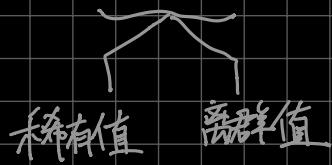
△ 梯度剪裁用于数据中包含许多异常值，会

造成梯度爆炸，梯度剪裁能很好地解决此

问题。
缺：使用者难以找到最佳学习率

How to use \Rightarrow 在特征工程中进行数据清洗时，若删除、替换了

异常值，并设置了合理的学习率，则并不需要梯度剪裁。



若网络偶尔发生错误（个别异常值未处理而产生），再打开梯度剪裁。

④ 隐藏层选择合适的激活函数。

问题

sigmoid. tanh : 梯度消失

Relu : 神经元死亡 \Rightarrow 受不良梯度影响

Leaky Relu. Elu. Selu : 高级激活函数，
一般可避免上述问题。

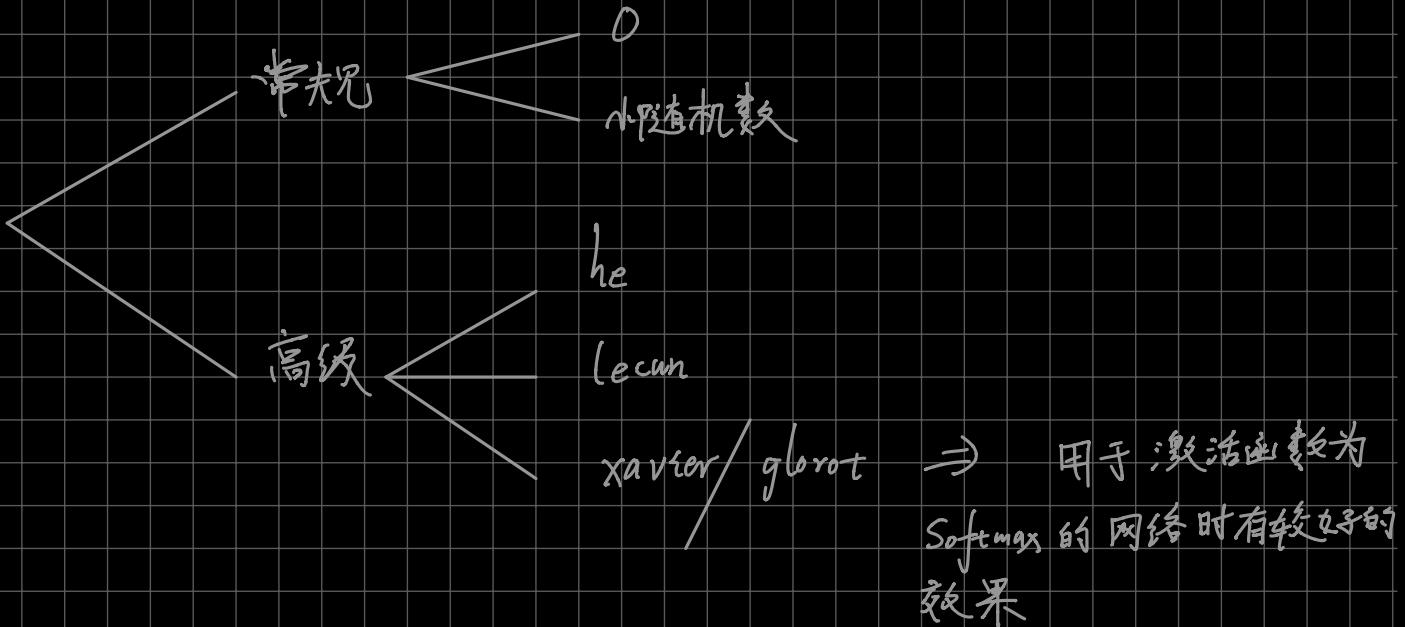
△ Softmax: 用于输出层激活

Linear : 不用 / 少用

回归问题一般输出层无需激活函数

△ 当 epoch 之间 训练误差 不会变化，可能
为 Relu 导致所有神经元已经死亡

⑤ 正确地初始化神经网络权重



How: 从高级中任选一个(此三者在几乎任何情况下效果均不错), 此后神经网络调节时再寻找最适合任务的权重初始化方式。

⑥ 正确地选择网络深度, 并非越深越好。

(i) 加深网络 在模型欠拟合时 能显著提高准确率。
参数规模 < 数据集规模

(ii) 若浅层网络没有学到数据特征(几乎没有效果), 那么加深网络也不会有效果。

(iii) 刚开始先用浅层网络(3-8层)进行训练, 当有效果时再逐步加大模型深度。⇒ 原因为训练速度慢, 能更快完成模型推理. 尝试不同结构, 快速收敛。

速调整 走参数。

①

②

③

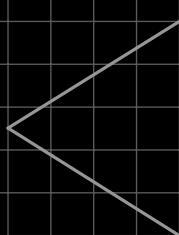


多。

①, ②, ③ 对模型准确率的影响将比模型深度大得

⑦ 正确地选择网络宽度(即隐藏层神经元数量)

过小：表达能力差



过大：训练速度慢，残留噪声难以消除。