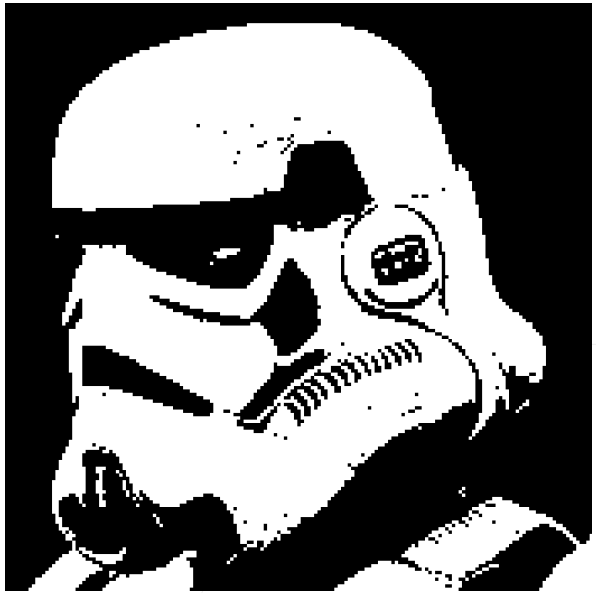
丢弃法 用于全连接层

另一种正则化方法，可能比权重衰退效果更好

动机

- 一个好的模型需要对输入数据的扰动鲁棒 即类似图片花一点也知道这是什么
  - 使用有噪音的数据等价于 Tikhonov 正则
  - 丢弃法：在层之间加入噪音

不断地随机加噪音，不同于之前的固定噪音



无偏差的加入噪音

- 对 $\mathbf{x}$ 加入噪音得到 $\mathbf{x}'$，我们希望

$$\mathbf{E}[\mathbf{x}'] = \mathbf{x}$$

- 丢弃法对每个元素进行如下扰动

$$x_i' = \begin{cases} 0 & \text{with probablity } p \\ \frac{x_i}{1-p} & \text{otherise} \end{cases}$$

即有p的概率变为0

有1-p的概率变为这个

=» 期望还是xi

使用丢弃法

- 通常将丢弃法作用在隐藏全连接层的输出上

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$ 线性+激活

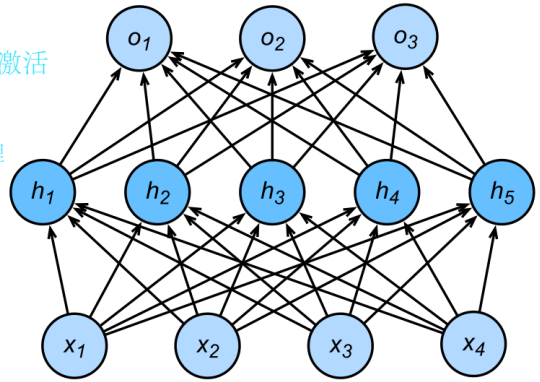
$$\mathbf{h}' = \text{dropout}(\mathbf{h})$$ 作前页处理

$$\mathbf{o} = \mathbf{W}_2\mathbf{h}' + \mathbf{b}_2$$

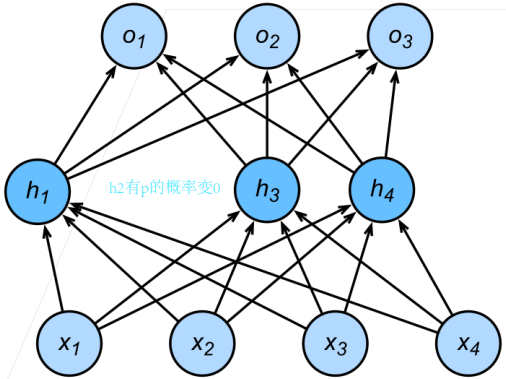
$$\mathbf{y} = \text{softmax}(\mathbf{o})$$

非线性归一化

MLP with one hidden layer



Hidden layer after dropout



可以理解为降低模型的路径依赖，每次训练时随机屏蔽几条路

推理中的丢弃法

- 正则项只在训练中使用：他们影响模型参数的更新
- 在推理过程中，丢弃法直接返回输入 此时不用更新模型

  预测

$$\mathbf{h} = \text{dropout}(\mathbf{h})$$ 即dropout后无变化

  - 这样也能保证确定性的输出

总结

- 丢弃法将一些输出项随机置0来控制模型复杂度
- 常作用在多层感知机的隐藏层输出上
- 丢弃概率是控制模型复杂度的超参数

  会造成收敛变慢，但是不需要改变learning rate，因为期望不变