手写数字识别的DL视角

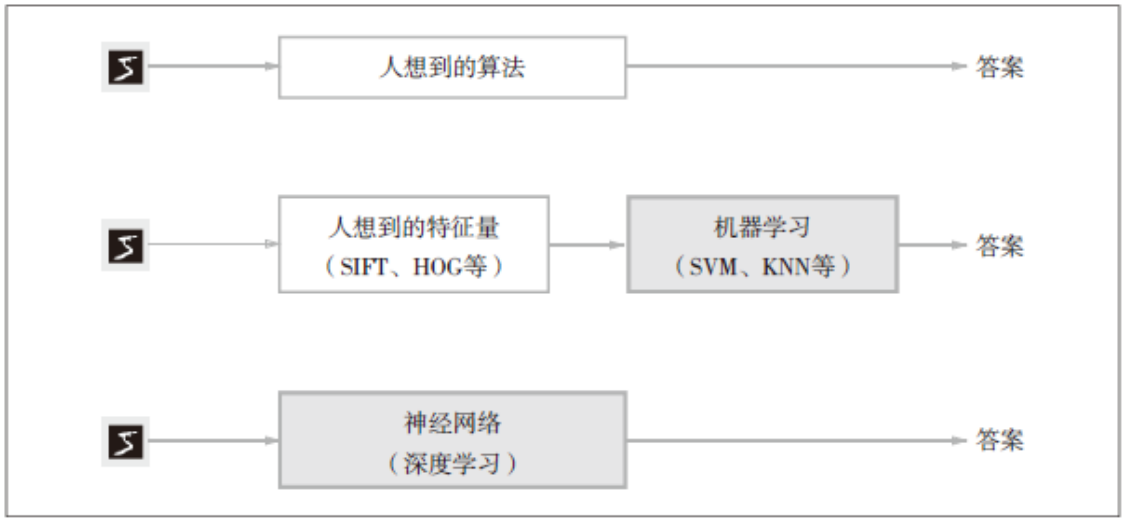

图4-2　从人工设计规则转变为由机器从数据中学习：没有人为介入的方块用灰色表示

神经网络的优点是对所有的问题都可以用同样的流程来解决。比如，不管要求解的问题是识别5，还是识别狗，抑或是识别人脸，神经网络都是通过不断地学习所提供的数据，尝试发现待求解的问题的模式。也就是说，与待处理的问题无关，神经网络可以将数据直接作为原始数据，进行"端对端"的学习。

泛化能力 -- 是指处理未被观察过的数据的能力

过拟合 -- 可以顺利地处理某个数据集，但**无法**处理其他数据集的情况

损失函数

- 均方误差MSE -- $E = 0.5\sum_k (y_k - t_k)^2$ -- 特性：考虑所有给出的预测结果 -> 适合**回归**
- 交叉熵误差CEE -- $E = -\sum_k t_k \ln y_k$ -- 特性：仅由正确解标签对应的预测结果决定 -> 适合**分类**

Mini-Batch学习

对于大规模的训练集，将其拆分为一个个Batch，喂入神经网络进行学习

选取方式：随机选取、顺序选取、有偏移的窗口选取

EPOCH -- 对于10000笔训练数据，用100的BATCH_SIZE，需要重复SGD100 次，所有的训练数据就都被"看过"了。此时，EPOCH=100。

梯度与导数

负梯度方向是梯度法（多用梯度下降GD）中变量的更新方向（损失函数减小最多的方向）

梯度场

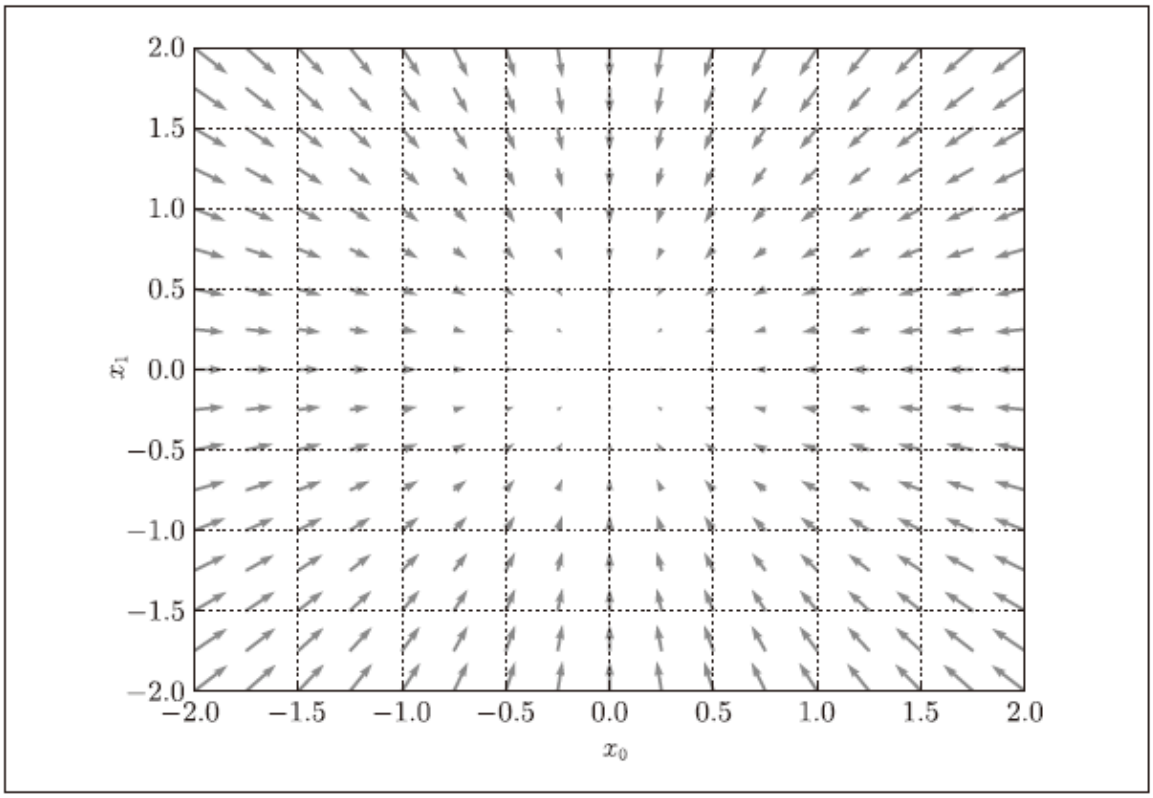

图4-9　$f(x_0, x_1) = x_0^2 + x_1^2$ 的梯度

基于GD的数值更新

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}$$

$$\frac{\partial L}{\partial \boldsymbol{W}} = \begin{pmatrix} \frac{\partial L}{\partial w_{11}} & \frac{\partial L}{\partial w_{12}} & \frac{\partial L}{\partial w_{13}} \\ \frac{\partial L}{\partial w_{21}} & \frac{\partial L}{\partial w_{22}} & \frac{\partial L}{\partial w_{23}} \end{pmatrix}$$

$$x_0 = x_0 - \eta \frac{\partial f}{\partial x_0}$$

$$x_1 = x_1 - \eta \frac{\partial f}{\partial x_1}$$

η：学习率，决定在一次学习中，应该学习多少，以及在多大程度上更新参数

学习率是一个超参数（无法学得，需要人工指定的参数），它的选取对训练的效果非常重要

np.nditer

提供了一个灵活的迭代器，可以无需使用n重for循环来遍历n维array

np.argmax

取出最大值对应的索引，根据axis灵活化

总结 - 最简单的人工神经网络的学习过程

- Step1 - 构建网络
- Step2 - 喂入数据（Mini-Batch思想）
- Step3 - 计算梯度
- Step4 - 更新参数（例如用SGD来minimize损失函数）

- Step5 - 重复，直到满意

随机梯度下降法SGD

对**随机选择的数据**进行的梯度下降法（不是对随机梯度位置进行下降）