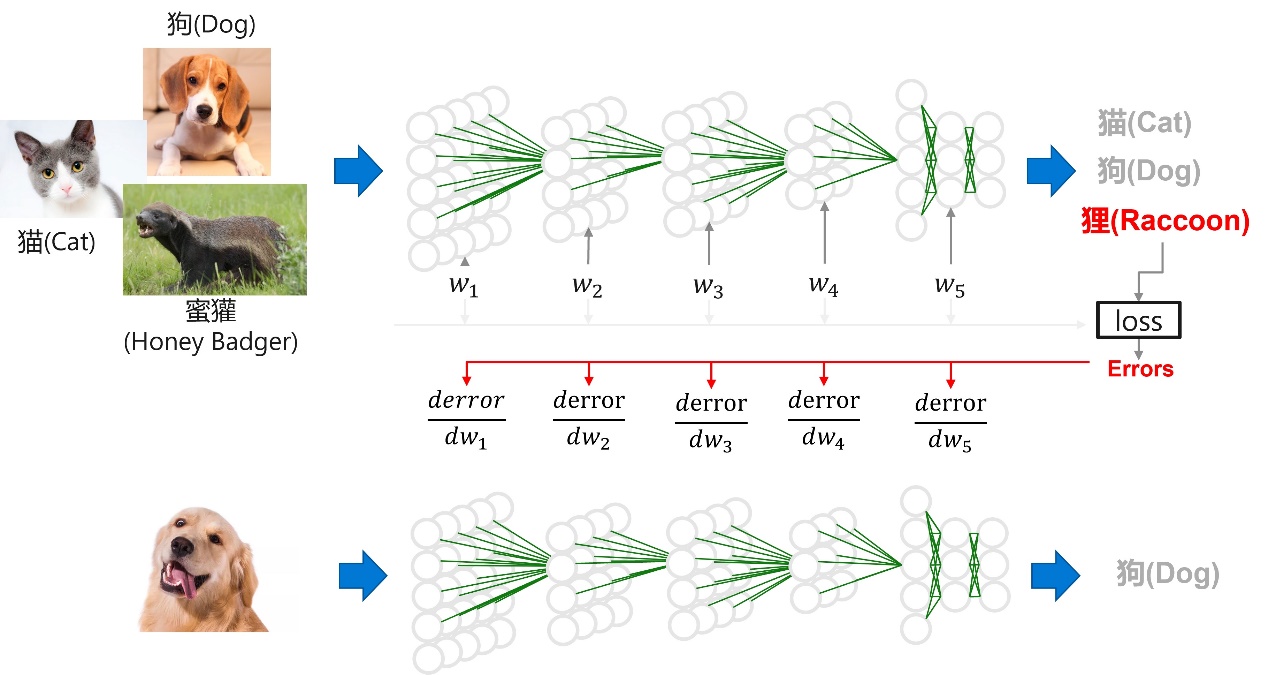
**赛博炼丹**



1. Sample：输入特征（Feature）与输出标签（Label）
2. Model：设计与开发模型结构
   * 卷积神经网络（Convolutional Neural Network）：由卷积层（Convolution Layer），池化层（Pooling Layer），全连接层（Fully Connected Layer）等算子（Operator）的组合形成
   * 循环神经网络（Recurrent Neural Network）：以长短时记忆（LSTM）等基本算子组合。【适合时序数据预测】
   * Transformer：自然语言处理
3. Training：根据优化算法搜索预测效果最好的权重。
   * Forward Propagation：由输入到输出传递信号。
   * Back Propagation：由输出到输入传递误差。
   * Weight Update：通过梯度下降法更新模型权重。
4. Inference：实际预测

开源框架：PyTorch、TensorFlow、Keras

卷积层：中间层就是自变量域的算子变换（函数空间上的映射），提取特征差异。

激活函数：引入非线性函数（曲线）来划分解空间。

池化层：保留主要特征，压缩数据大小。

全连接层：局部特征 --整合--> 全局特征，用来计算总得分。

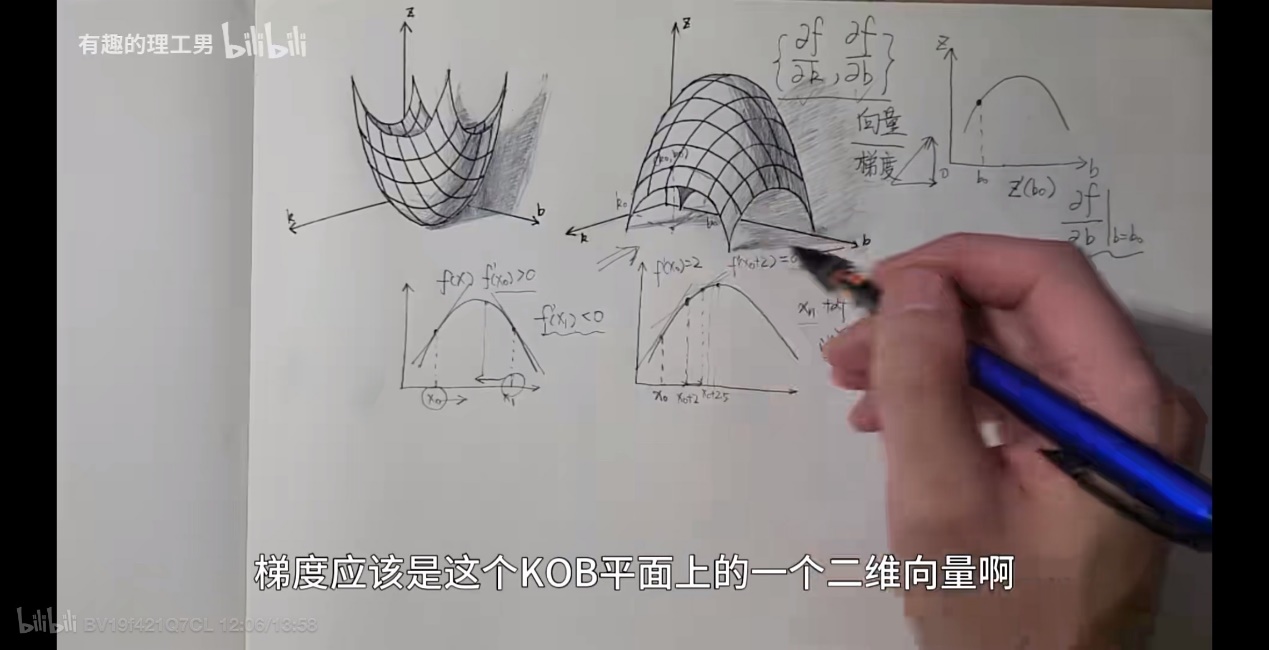
学习结果：数学模型中的方程系数。梯度下降法就是逼近系数的一种方法。

**数学概念**

**导数** = lim (Δy/Δx)，变化率

**方向导数：**多元函数而言，某个射线方向上的导数

**偏导数：**坐标轴方向上的方向导数，其他方向的方向导数为偏导数的合成



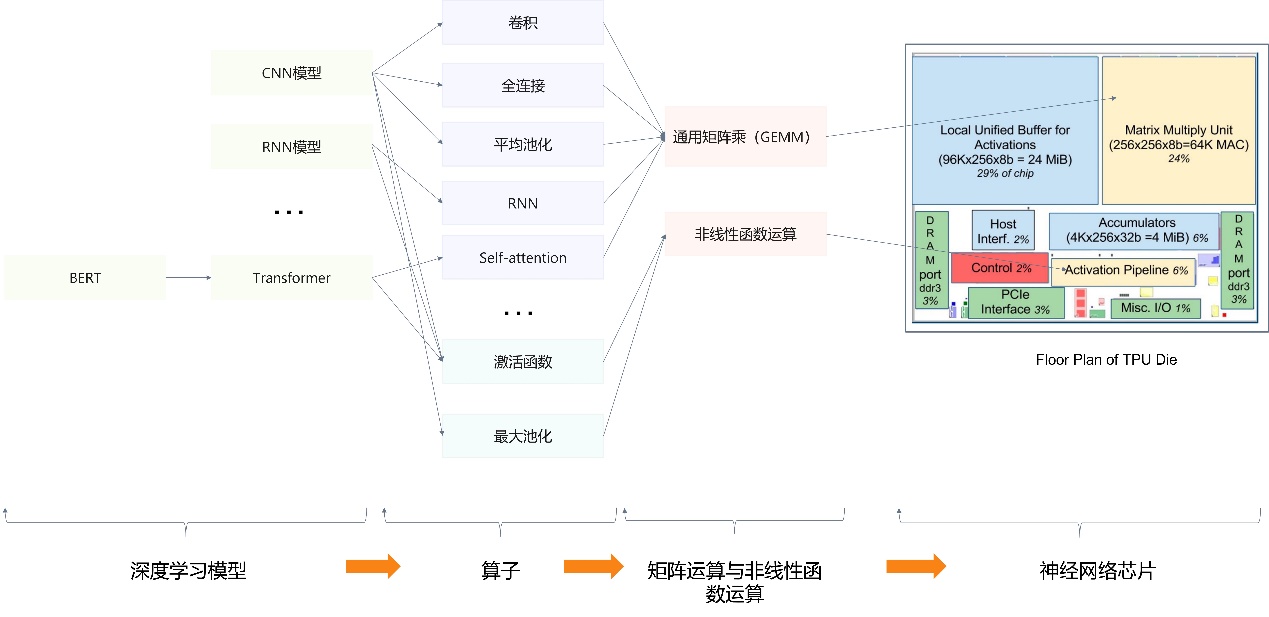
**梯度：**所有坐标轴的偏导数所构成的向量。在变量空间的某一个点处，沿着梯度方向的方向导数最大，并且方向导数最大值为梯度的长度。（与等高线垂直）

**梯度下降：**逆着梯度方向，寻求极值

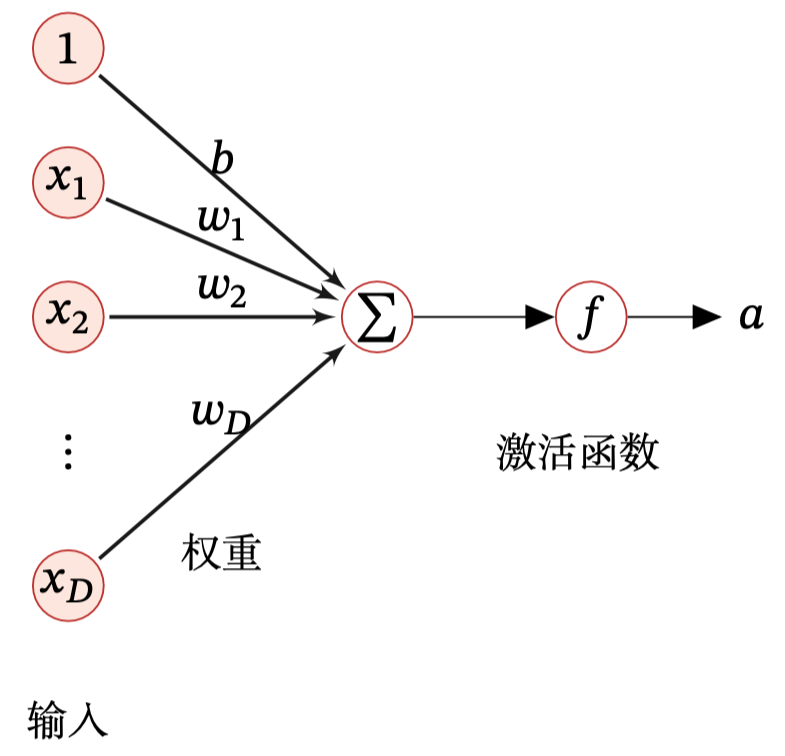
**通用近似定理**（Universal Approximation Theorem）：前馈神经网络（Universal Function Approximators）能近似任意函数，且被近似的目标函数通常为输入输出都在欧几里得空间【余弦定理相似性的前提】的连续函数。

**卷积(Convolution)：**一种信号变换算子（滤波器），提取特征。

**线性代数**：大部分的深度学习算子可以抽象为线性代数运算，如通用矩阵乘（GEMM）。



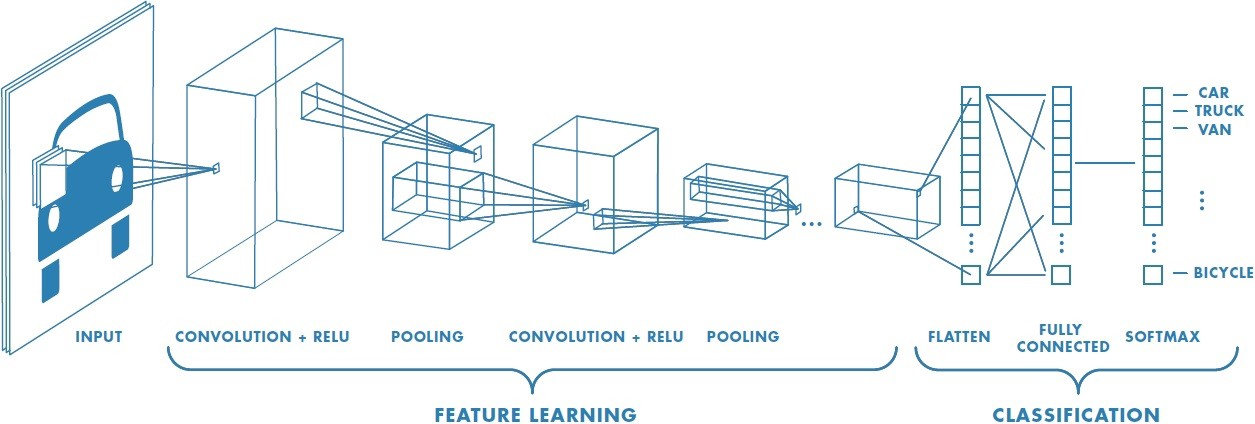
**神经元**



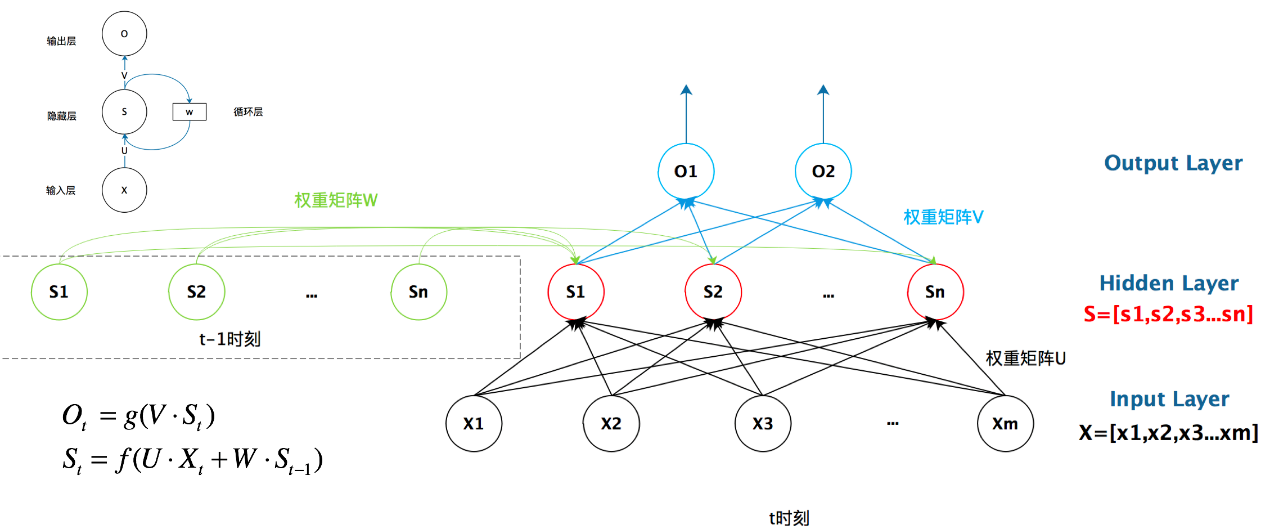
一个没有激活函数的神经网络将只不过是一个线性回归模型罢了，不能解决现实世界中的大多数非线性问题。

（输入，权重） -> 求和、偏移 -> （激活函数，输出）

**CNN**



**RNN**



**Transformer**

1. 分词（tokenization）：词频统计表压缩信息量

2. 向量化（embedding）：文本段落to数值向量

3. 上下文（attention）：从多个语义角度去计算相关性概率

4. 整合思考（）

5. 后置优化

|  |  |
| --- | --- |
| Prompt | 提示词工程：立角色、述问题、定目标、补要求 |
| Agent | 智能体：调API |
| RAG | 检索增强：知识库的搜索结果附加在提问中 |
| Fine Tuning | 微调：人工经验的数据调参 |