Materials on machine learning

for 3rd CMS China Winter Camp

主讲：李聪乔

第三届中国CMS冬令营 · 北京航空航天大学

19 January, 2025

Outline

This tutorial will cover

- ➔ Very brief introduction of BDT/DNN basics
- ➔ Practical aspects in BDT/DNN training
- ➔ **Hands-on session:**

https://github.com/colizz/ml-tutorial/tree/v2025-01-cc3

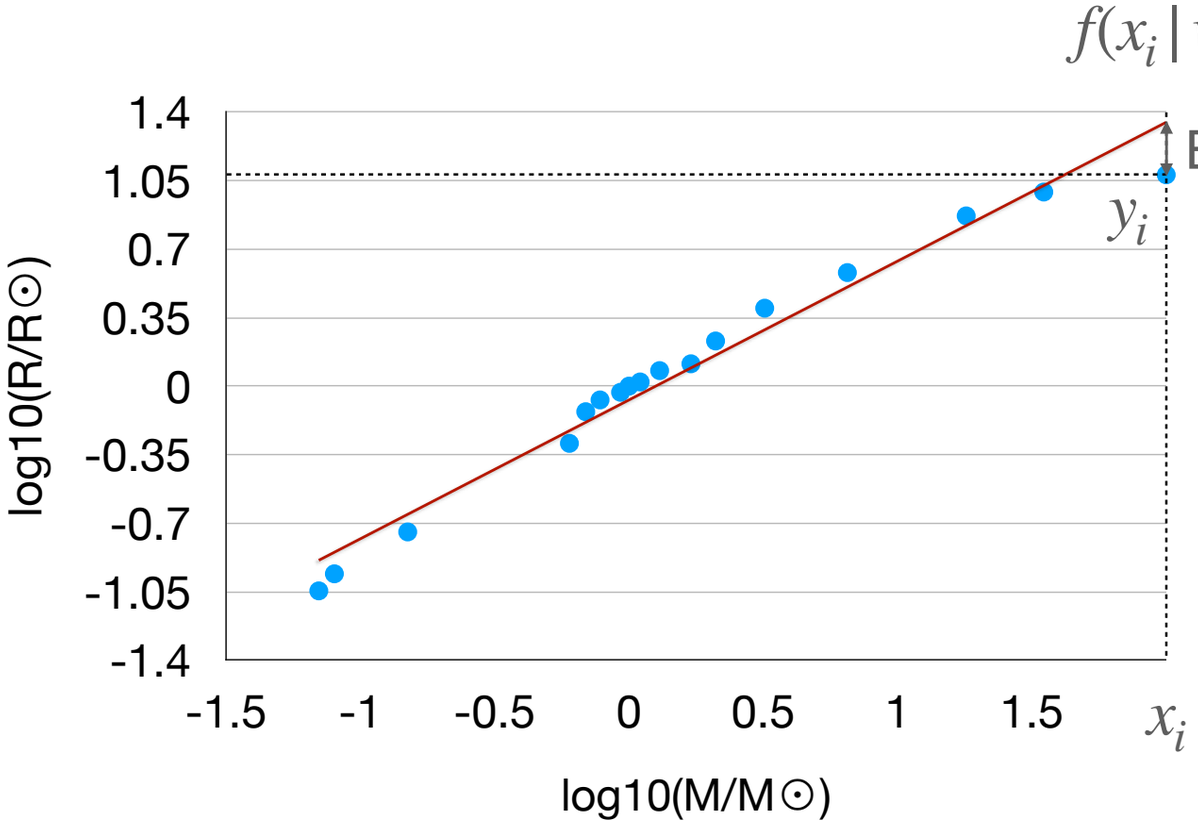
References:

1. CMS DAS material
   https://indico.cern.ch/event/1462056/contributions/6155284/attachments/2936806/5276739/ml_das_13_01_25_intro.pdf
   https://indico.cern.ch/event/966368/contributions/4172531/attachments/2168986/3662057/CMSDAS2021_ML_WrapUp_11Jan2021.pdf
2. Early Winter Camp materials
3. Introduction to Transformers (w/ hands-on practices on ParT/ParticleNet): https://github.com/colizz/ml-tutorial/tree/v2025-01-nku

Machine Learning?

机器学习的理解：“数据驱动”的算法，基于样本数据来进行预测或者做出判断，其规则并不显式写入程序之内。

例如：**线性回归** 就是一种机器学习算法



- Linear model:

$$f(x \mid w) = w^\intercal x \quad (w \in \mathbb{R}^{D+1})$$

- How do we select the parameters w?
  - We want $y_i \approx f(x_i \mid w)$
- Squared loss: $L(y, y') = (y - y')^2$

(Least squares)

Learning objective: $\arg\min_w \sum_{i=1}^{N} L(y_i, f(x_i \mid w)) = \arg\min_w \sum_{i=1}^{N} (y_i - w^\intercal x_i)^2$

最小化目标函数

Typical tasks of machine learning

X

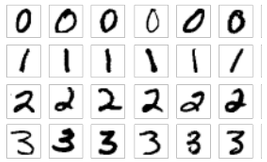
Model with learnable parameters

$$Y = f(X)$$

Y

分类

输入手写数字图像

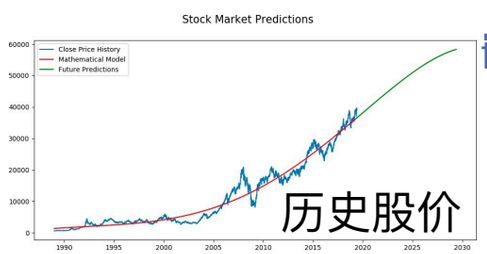


手写数字分类
10 种类别

输入图像

ImageNet 分类：
1000 种类别

回归



嵌入到输入矢量

下一时刻股价预测

Machine learning: general pipelines

- Training dataset: $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where $x \in \mathbb{R}^D$ and $y \in \mathbb{R}$
- Model / hypothesis class: $f(x \mid w) = w^\intercal x$ (linear models)
- Loss function: $L(y, y') = (y - y')^2$ (squared loss)
- Optimization algorithm to minimize the learning objective:

$$\arg\min_w \sum_{i=1}^{N} L(y_i, f(x_i \mid w))$$

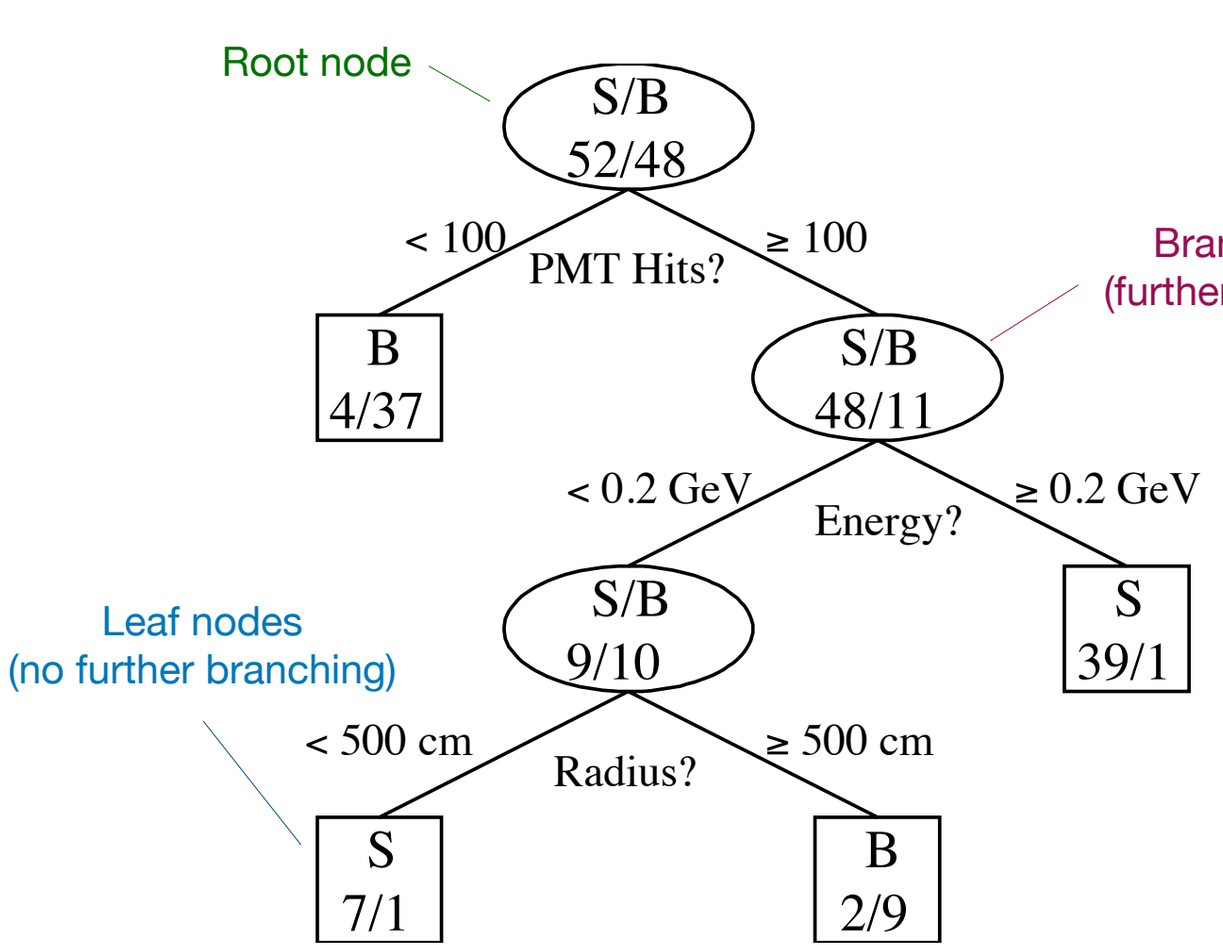
- Cross validation and model selection:

- Testing and deployment

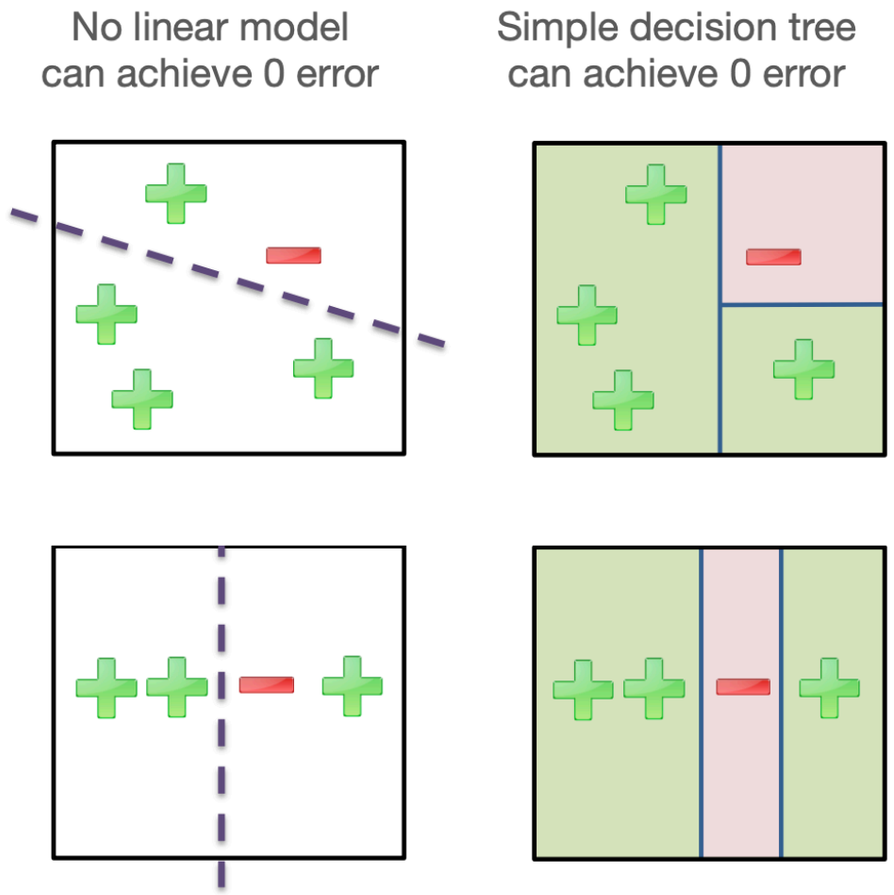
Important: if a testing set is available, never use it to make decisions on the model!

验证集决定选哪个模型，测试集用来最终部署

Decision trees



用多个变量进行决策



决策树：1) 非线性模型；2) 单个决策只用单一变量

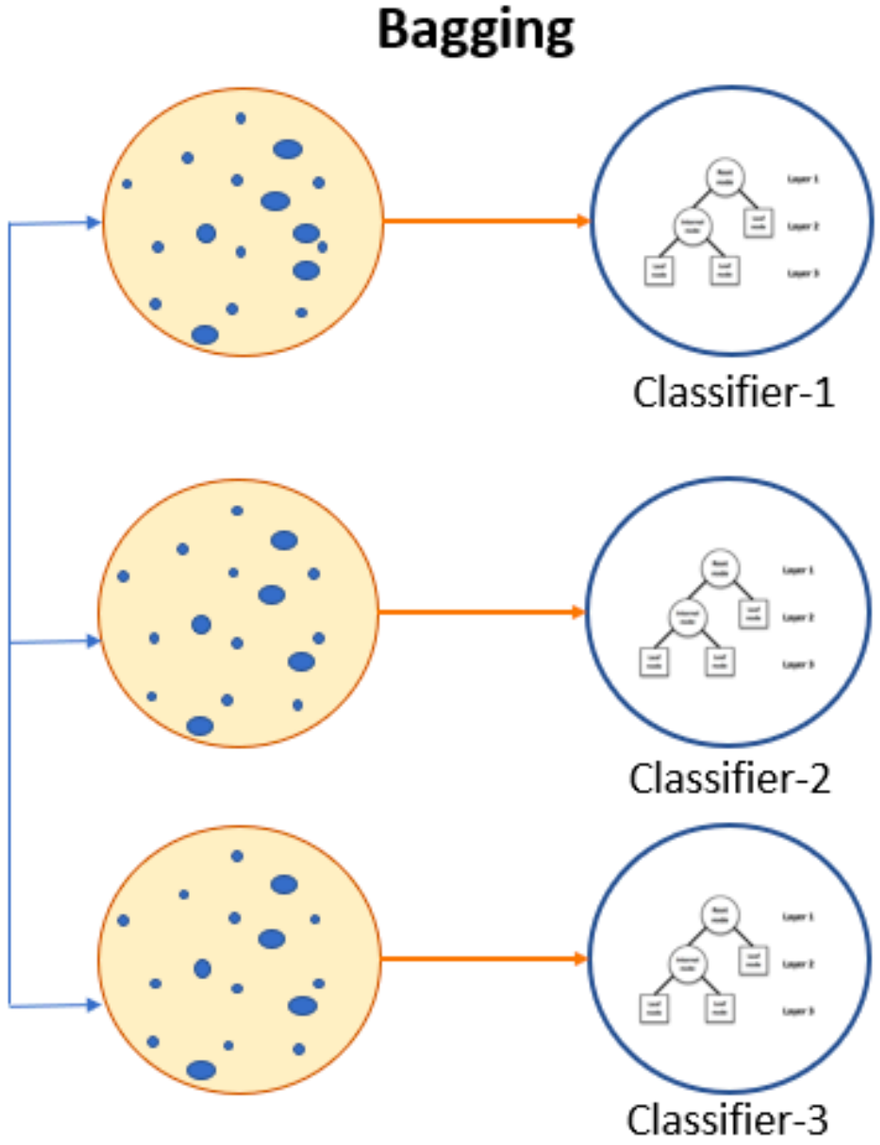
Boosted decision trees (BDT)

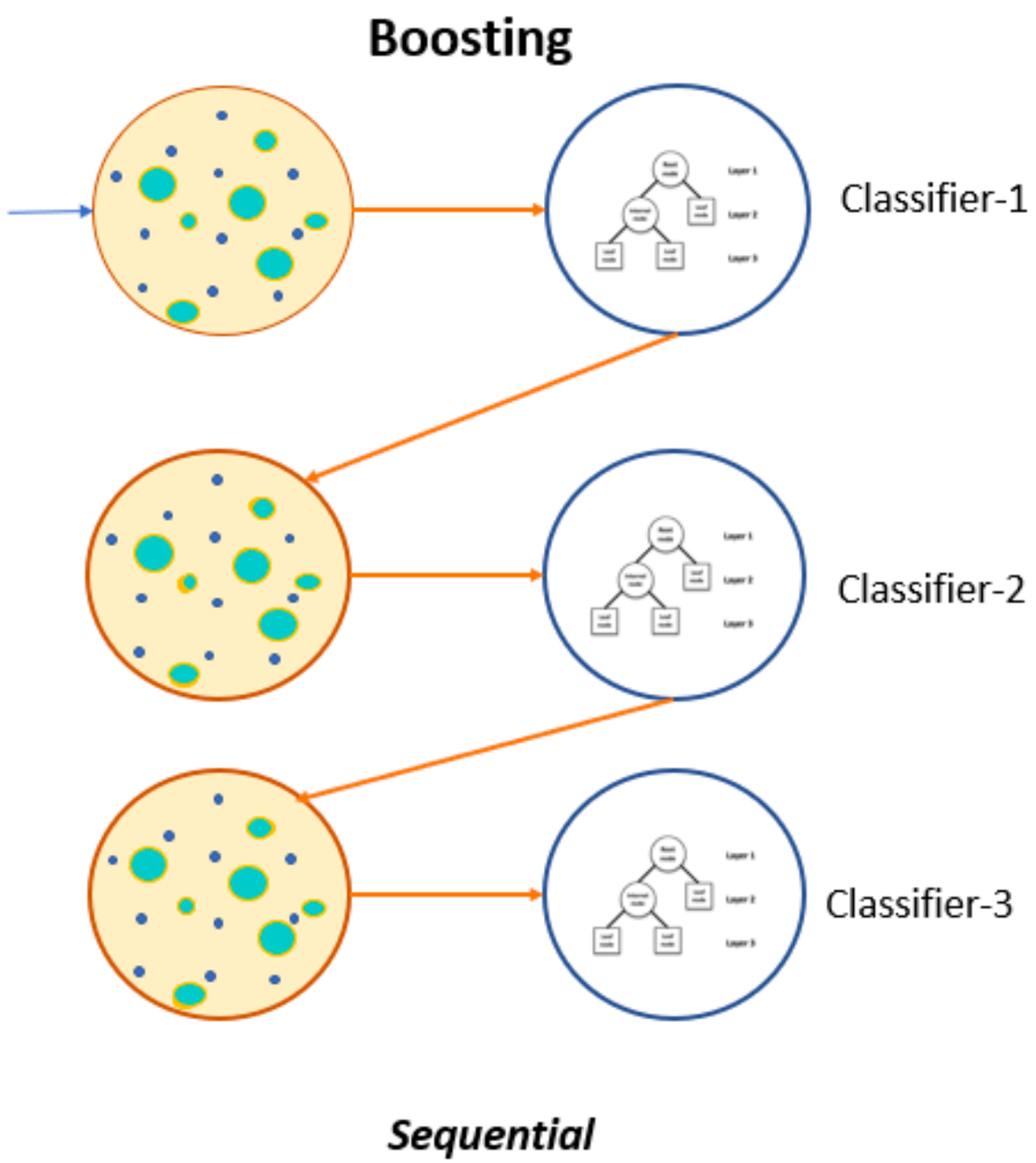
用多棵“决策树”提升性能

Bagging: 选不同数据子集，训练多个模型

Boosting: (高能物理常用)

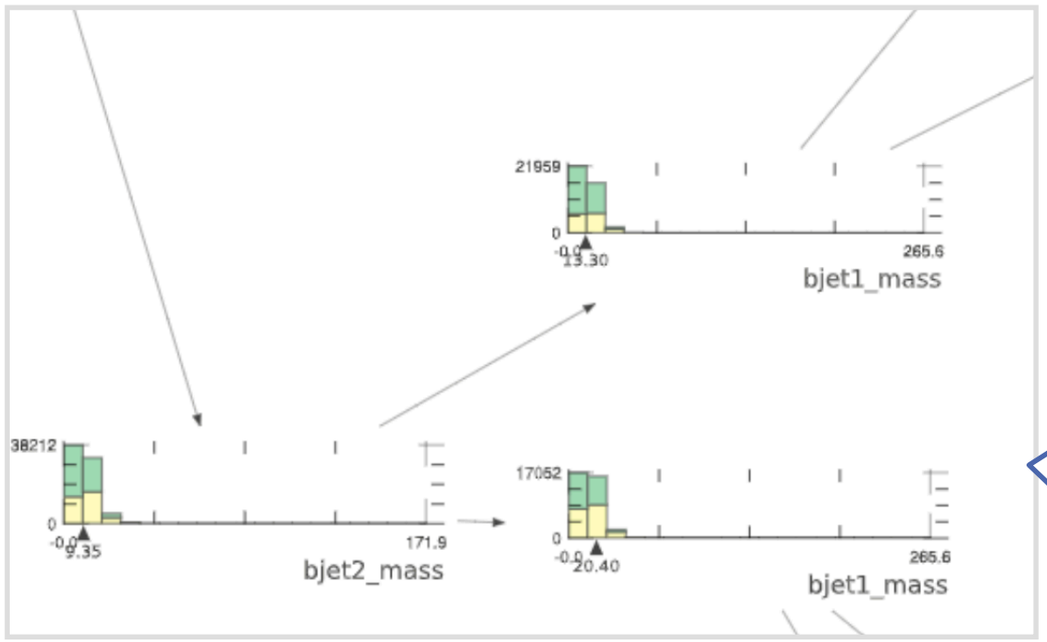
- 不断拟合新的决策树，作为之前决策残差
- 决策树的weight依赖于判断正确率

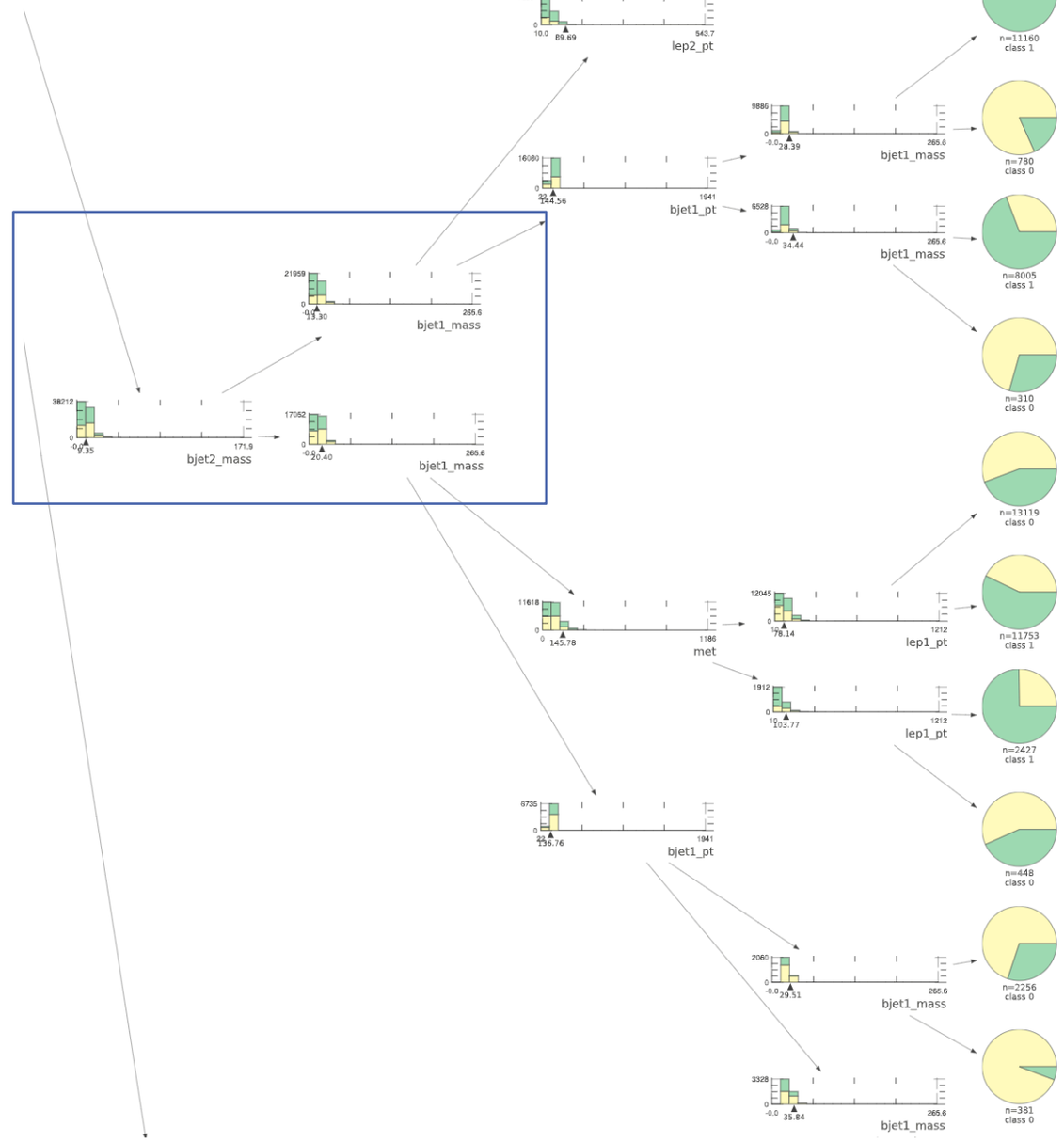




BDT: a visualization example

BDT 的可视化举例

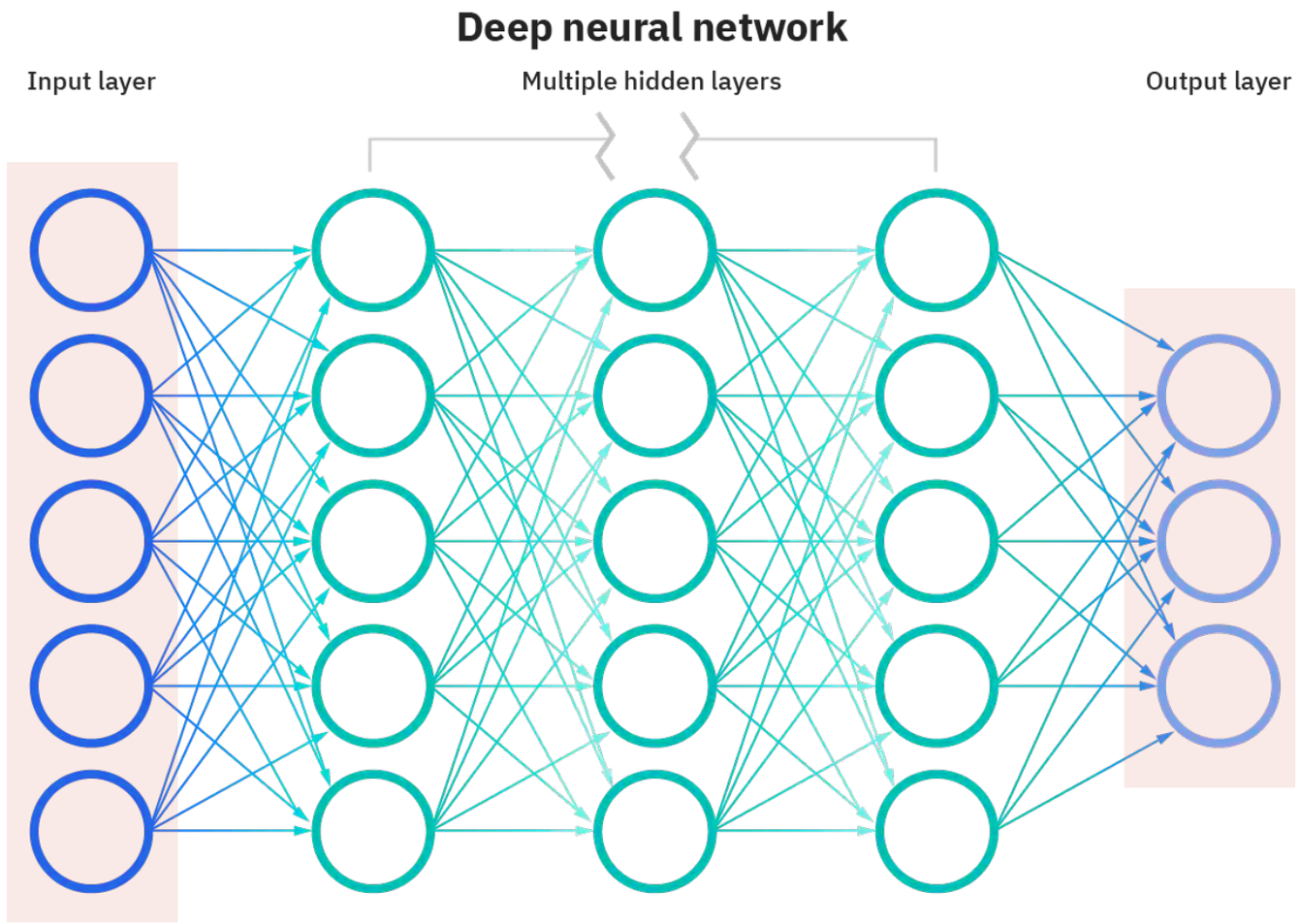




例如：

- 第一棵树决策后，认为是信号的，包含60%真实信号和40%假信号(本底)，则赋予权重 0.6。
- 在60%的这些信号中，希望后续的树拟合出残差0.4；在其余误判断的40%本底中，希望后续的树拟合出残差 -0.6.
- …
- 最后每一个分支的分数，是所有决策树的决策weight之和

Deep neural networks



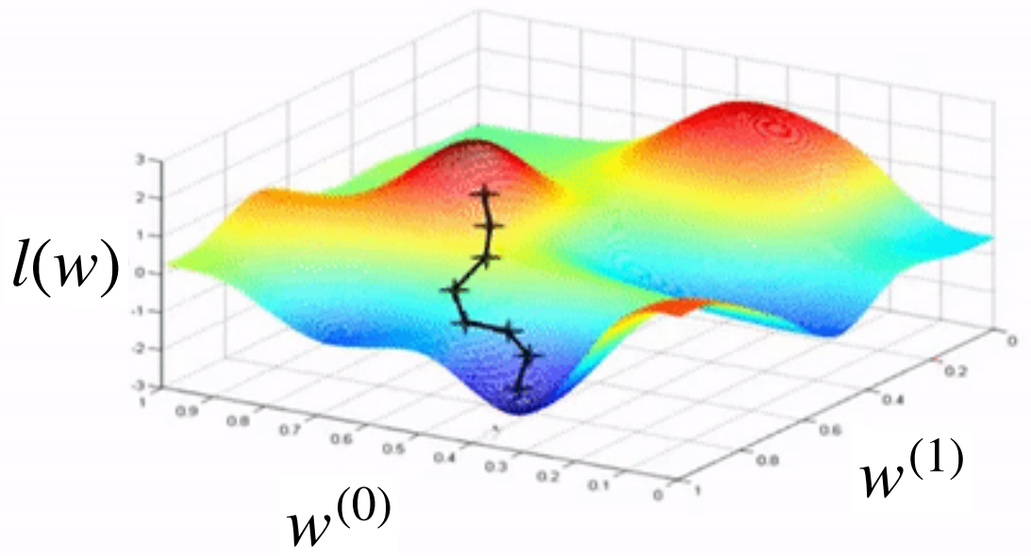
X $Y = f(X)$ Y

DNN optimization: gradient descent

- Set $w(t=0)$ to some values (e.g., $w(0)=0$ or some random value)
- At iteration t,
  - Compute the gradient $\nabla_w l(w(t))$: direction of steepest increase of $l(w)$ at $w(t)$
  - Take a small step in the opposite direction:

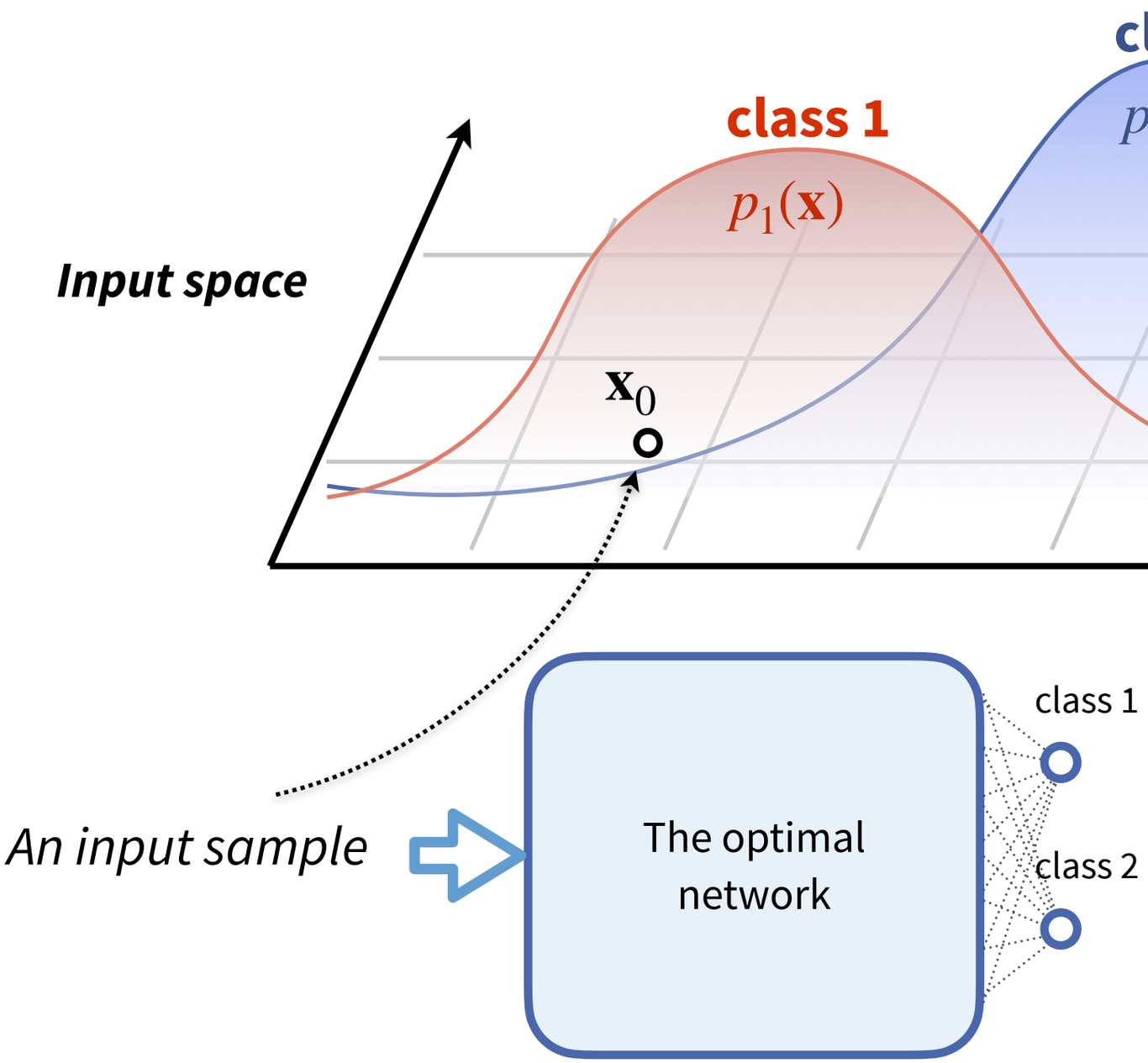
$$w(t+1) = w(t) - \eta \nabla_w l(w(t))$$

Step size / learning rate



Statistical principle behind DNN training

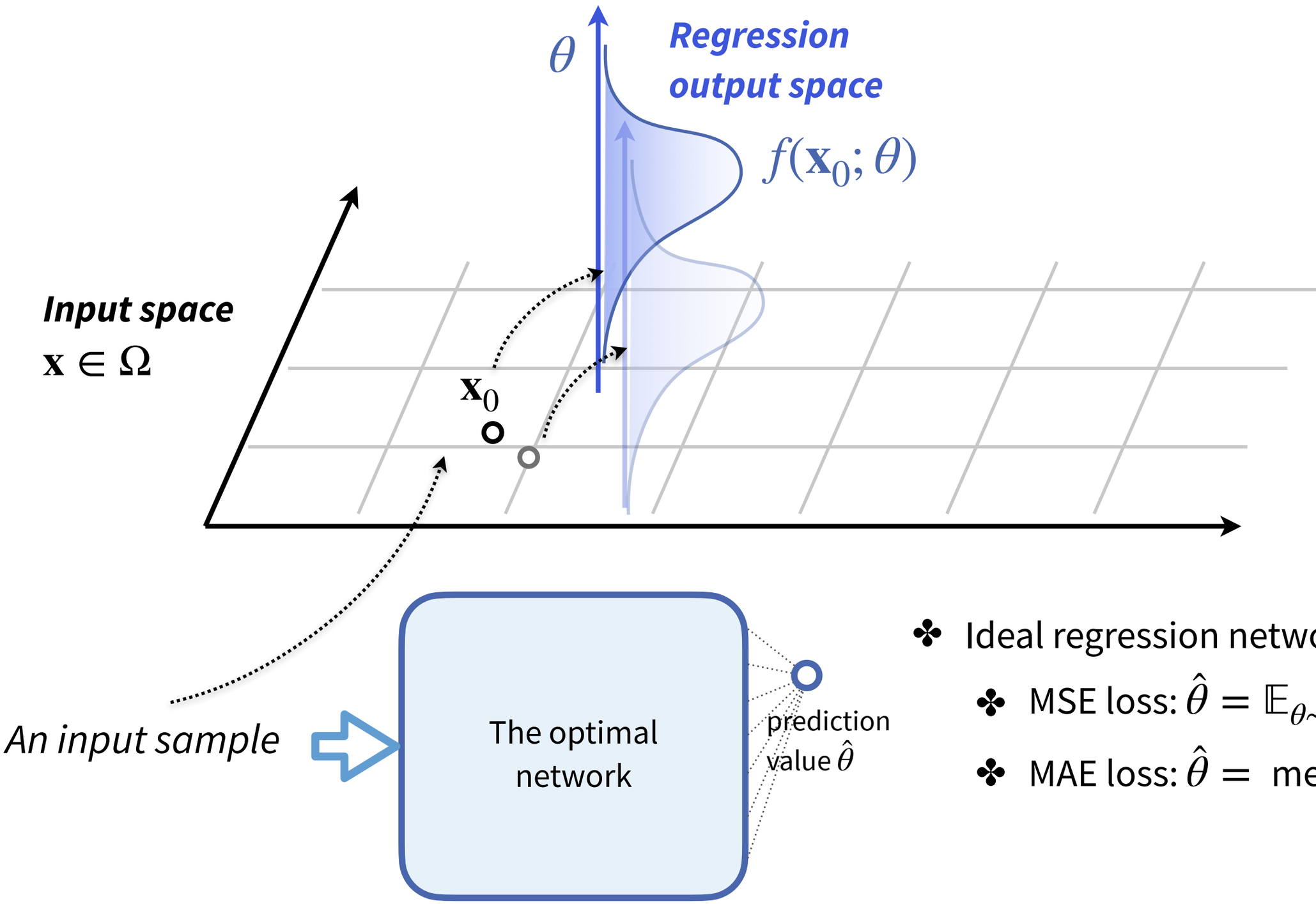
➔ Why optimizing cross-entropy (交叉熵) loss for classification tasks?



- Ideal classifier network results in
  $g_1 : g_2 : \ldots = p_1(\mathbf{x}_0) : p_2(\mathbf{x}_0) : \ldots$
- It is a direct estimation of p
- The **network capacity** decides how close the estimation is

Statistical principle behind DNN training

➔ Why optimizing MSE / MAE (均方差/平均绝对误差, or L2/L1) loss for regression tasks?



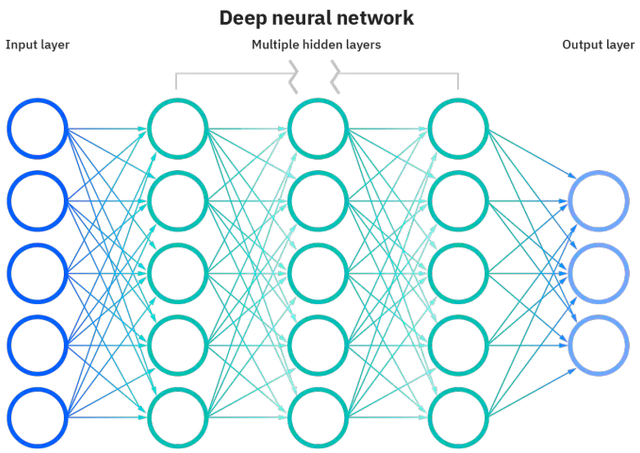
- Ideal regression network results in:
  - MSE loss: $\hat{\theta} = \mathbb{E}_{\theta \sim f(\mathbf{x};\theta)}\,\theta$ (mean!)
  - MAE loss: $\hat{\theta} =$ median

DNN input engineering

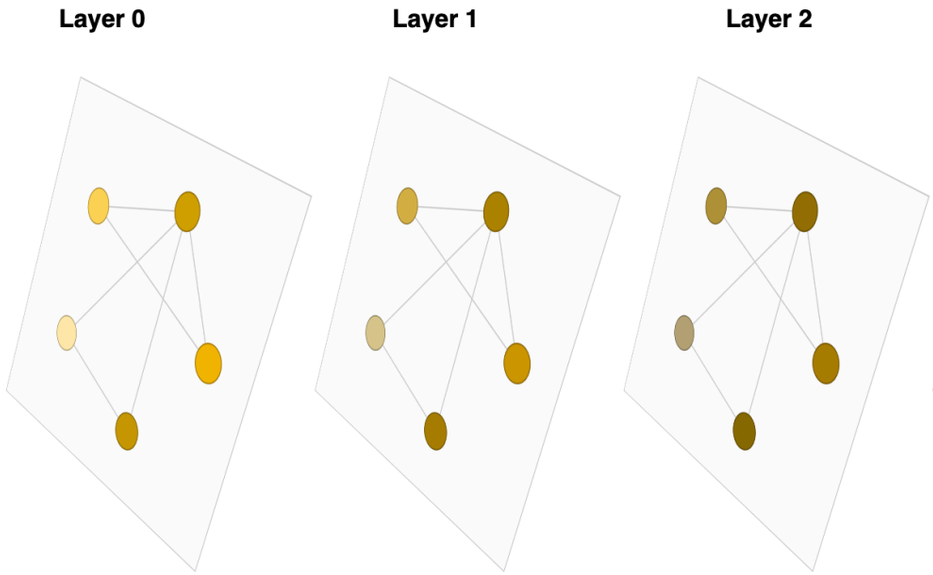
➔ 为了便于DNN优化，需要进行数据预处理：

- ❖ shift + scaling → 例如，使输入变量分布满足均值为0，方差为1
- ❖ 长尾分布 (e.g. jet p_T)，通过 log(|x|) 、log(|x|+1) 变换获得接近高斯的分布

➔ flattened 1D vector & tokenized input?



- 展开为1D vector 输入 DNN



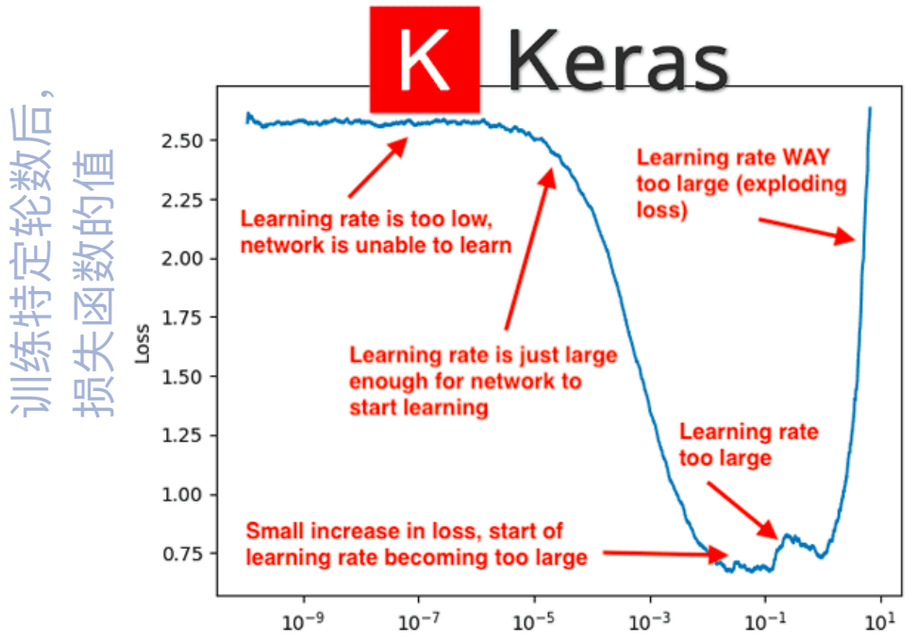
- 构造有层级的输入 (每个lepton / jet 等object当做一个 input node / token) → 尊重数据原本的属性
- 用更具表达性的 GNN / Transformer 来分析

见 hands-on 实例！

DNN hyperparameter tuning

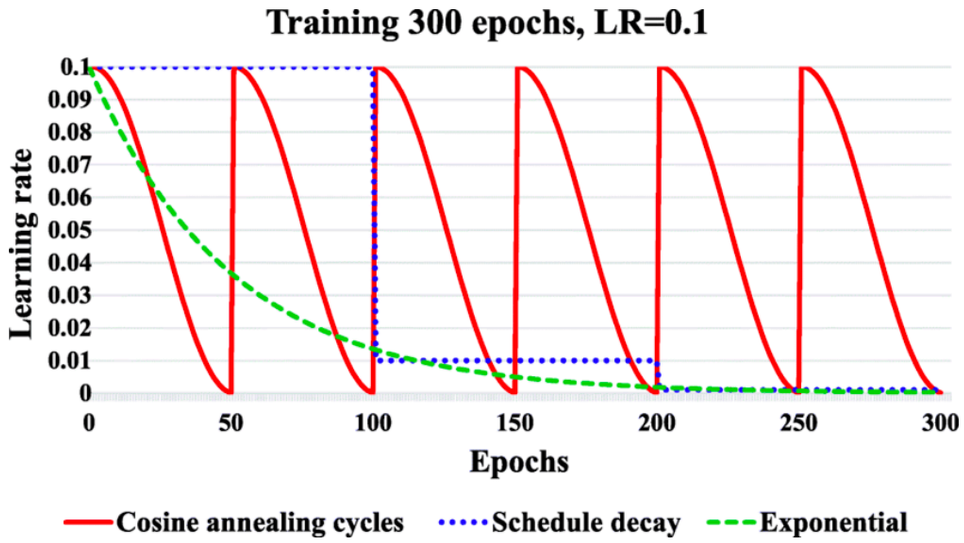
➔ 学习率 (learning rate)

- ❖ learning rate 是最重要的超参量
- ❖ 反映训练的每一步按照梯度更新参数的步长，可以以指数为跨度进行调节，最后挑选始终的值



➔ 学习率的衰减 (decay rate)

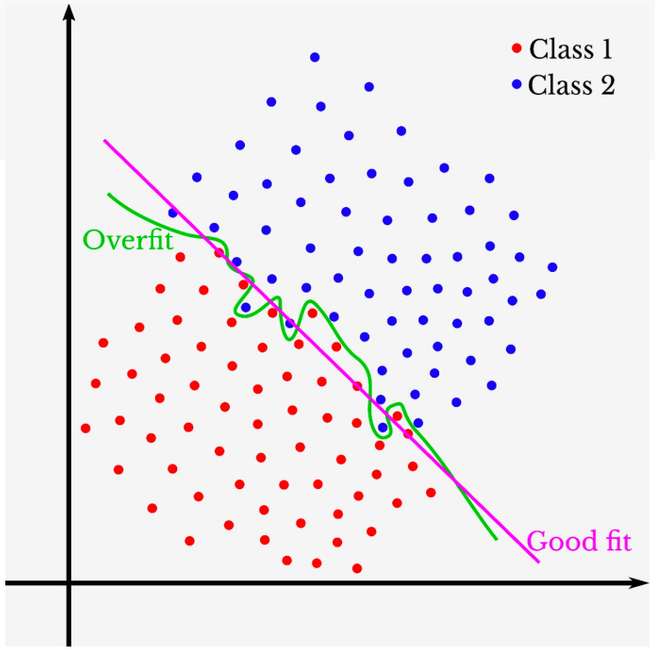
- ❖ 学习率选用何种模式衰减？阶梯式衰减、指数衰减、余弦退火衰减，在不同的场景下各有应用
- ❖ 对于基础的神经网络，选择简单的阶梯式或指数衰减即可

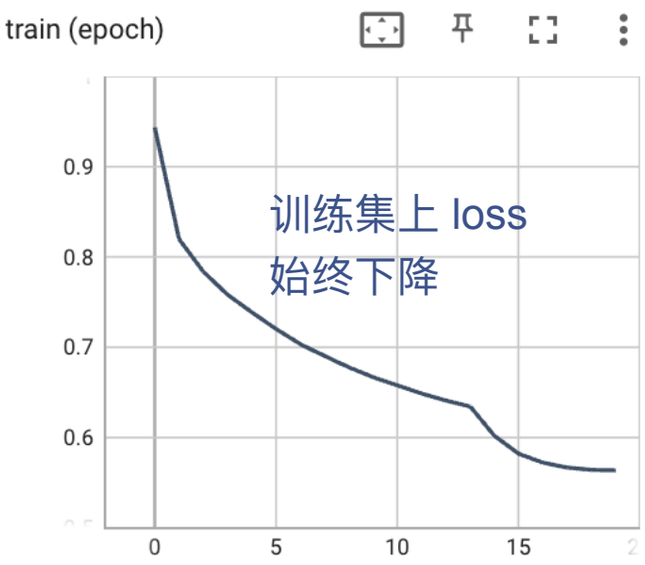


DNN: overfitting

➔ 过拟合

- 表现：loss 在训练样本上始终下降，但每轮 epoch 结束后在测试样本上运行发现不再下降，甚至有所上升





- 原因：
  - 模型设计过大，有很多冗余参数
  - 训练的样本量太小，无法支持深层神经网络进行训练（经验上，NN 比 BDT 先天需更多样本）
- 改进方案：
  - 减小模型尺寸（如减少层数、减小每层的大小）
  - 加 dropout：一种训练技巧，每批训练时随机删除一些神经元 → keras 里有相应设置
  - …

Let’s begin our hands-on session

Resources: “deep-learning” deep learning

➔ 计算机视觉课程：

- ❖ UMich EECS 498: Deep Learning for Computer Vision [🔗Youtube] [🔗bilibili]

➔ 经典文章导读系列：

- ❖ Bilibili: 跟李沐学AI [🔗link]

Resources within CMS

- **CMS ML Group** (https://twiki.cern.ch/twiki/bin/viewauth/CMS/CMSMachineLearning)
  - Hosts the bi-weekly CMS ML Forum: https://indico.cern.ch/category/12412/
  - Journal club: https://cms-ml-journalclub.web.cern.ch/

- **CERN Inter-Experiment LHC Machine Learning Working Group** (https://iml.web.cern.ch/)
  - HEP ML Resources: https://github.com/iml-wg/HEP-ML-Resources
  - HEP ML Living Review: https://github.com/iml-wg/HEPML-LivingReview