风电大数据集——框架——算法——输出：多元可视化技术展现

使用深度学习而非机器学习的原因：

机器学习是实现人工智能的必经路径，深度学习是一种实现机器学习的技术。机器学习能够适应各种数据量，特别是数据量较小的场景。但如果数据量迅速增加，那么深度学习的效果将更加突出，这是因为深度学习算法需要大量的训练数据才能展现出神奇的效果。因此产生大量实时数据和历时数据的风电系统需要使用深度学习。

风电的数据从何而来

依托国网项目《基于深度学习的短期风电功率预测在线建模》，从D5000平台获取实测功率和实时气象数据及其他相关运行数据，从气象部门获取数值天气预报数据等已有系统的数据采集交互。

使用PyTorch/TensorFlow框架的原因：

深度学习的主流框架有，Tensorflow，PyTorch，Caffe，Keras以及MXNet等，其中Tensorflow是工业界主流框架，Pytorch由于其灵活性在科研中得到了广泛使用，且自2017年问世以来越来越来越流行。

TensorFlow在GPU的分布式计算上更为出色，在数据量巨大时效率比pytorch要高一些。

PyTorch的实现方式完全和Python的语法一致，简洁直观；动态计算图的设计思想正被越来越多人所接受，PyTorch已成为动态图框架的代表。PyTorch的发布让许多用户第一次发现原来深度学习框架可以如此灵活、如此容易、还如此快速。这得益于思想直观明了的动态图，它更符合人的思考过程，可以任意修改前向传播，还可以随时查看变量的值。动态执行，可以交互式查看修改，可在Python和Jupyter Notebook上随时查看和修改变量，十分灵活。

动态图带来的另外一个优势是调试更容易，在PyTorch中代码报错的地方往往就是写错代码的地方，而静态图需要先根据你的代码生成Graph对象，然后在session.run ()时报错，这种报错几乎很难找到对应的代码中真正错误的地方。

不选择其他框架的原因：Caffe灵活性缺失、扩展难、依赖众多环境难以配置、应用局限在图像处理领域，尤其是CNN上效果确实非常好，但是对于一些文本、流数据、时间数据的处理效果不好；Keras虽然上手容易但层层封装使其速度过慢且用户使用Keras主要是在调用接口，很难真正学到深度学习的内容；MXNet在同样的模型下能占用更小的内存和显存，并且在分布式环境下，MXNet展现出了明显优于其他框架的扩展性能，但它普及程度不足且接口文档不够完善。

在深度学习中使用分布式随机梯度下降算法的原因：

深度学习的架构和最新发展，包括CNN、RNN、GAN，都离不开梯度下降算法。梯度下降算法的最终目的，是找到全局最小值。

普通的批量梯度下降算法（BGD）是每次迭代把所有样本都过一遍，每训练一组样本就把梯度更新一次。随机梯度下降算法（SGD）是从训练样本中随机抽出一组，训练后按梯度更新一次，然后再抽取一组，再更新一次，在样本量及其大的情况下，可能不用训练完所有的样本就可以获得一个损失值在可接受范围之内的模型。

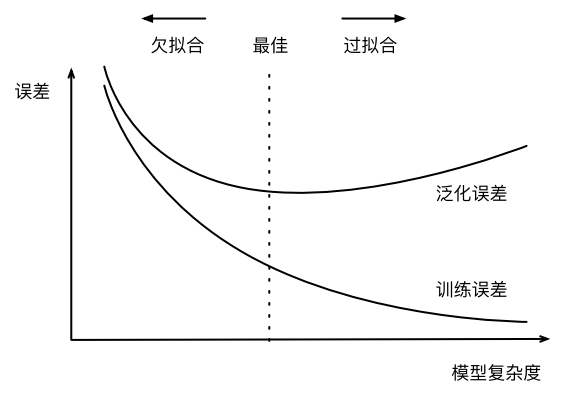
难点：

1过拟合

当模型在训练数据集上准确时，它在测试数据集上却不⼀定更准确，这是因为存在着训练误差（模型在训练数据集上表现出的误差）和泛化误差（模型在任意⼀个测试数据样本上表现出的误差的期望，并常常通过测试数据集上的误差来近似）。

欠拟合：模型⽆法得到较低的训练误差，常出现于复杂度过低的模型。

过拟合：模型的训练误差远小于它在测试数据集上的误差，常出现于复杂度过高的模型。



解决过拟合的方法：

权重衰减（weight decay）：权重衰减等价于l2范数正则化（regularization）。正则化通过为模型损失函数添加惩罚项使学出的模型参数值较小，是应对过拟合的常⽤⼿段。

另有丢弃法(Dropout)常用于解决过拟合。

2.梯度消失/梯度爆炸

训练神经网络，尤其是深度神经所面临的一个问题就是梯度消失或梯度爆炸，也就是训练神经网络的时候，导数或坡度有时会变得非常大，或者非常小（甚至以指数方式变小），这会加大训练的难度。这是因为在深层网络中，由于网络过深，如果初始得到的梯度过小，或者传播途中在某一层上过小，则在之后的层上得到的梯度会越来越小，即产生了梯度消失。梯度爆炸也是同样的。一般地，不合理的初始化以及激活函数，如sigmoid等，都会导致梯度过大或者过小，从而引起消失/爆炸。

解决方法：

梯度剪切、正则

梯度剪切这个方案主要是针对梯度爆炸提出的，其思想是设置一个梯度剪切阈值，然后更新梯度的时候，如果梯度超过这个阈值，那么就将其强制限制在这个范围之内。这可以防止梯度爆炸。另外一种解决梯度爆炸的手段是采用权重正则化（weithts regularization）比较常见的是l1和l2正则。

ReLu、leakReLu等激活函数

ReLu：其函数的导数在正数部分是恒等于1，这样在深层网络中，在激活函数部分就不存在导致梯度过大或者过小的问题，缓解了梯度消失或者爆炸。同时也方便计算。当然，其也存在一些缺点，例如过滤到了负数部分，导致部分信息的丢失，输出的数据分布不在以0为中心，改变了数据分布。

leakrelu：为解决ReLu的0区间带来的影响，其数学表达为：leakrelu=max(k\*x,0)其中k是leak系数，一般选择0.01或者0.02，或者通过学习而来。

批规范化

批规范化通过规范化操作将输出信号x规范化到均值为0，方差为1保证网络的稳定性。

是深度学习发展以来提出的最重要的成果之一了，目前已经被广泛的应用到了各大网络中，具有加速网络收敛速度，提升训练稳定性的效果，Batch Normalization本质上是解决反向传播过程中的梯度问题。

有一些从 0 到 1 而不是从 1 到 1000 的特征值，通过归一化所有的输入特征值𝑥，以获得类似范围的值，可以加速学习。所以 Batch 归一化起的作用的原因，直观的一点就是，它在做类似的工作，但不仅仅对于这里的输入值，还有隐藏单元的值。

它可以使权重比你的网络更滞后或更深层，比如，第 10 层的权重更能经受得住变化。

残差结构

残差的方式，能使得深层的网络梯度通过跳级连接路径直接返回到浅层部分，使得网络无论多深都能将梯度进行有效的回传。

LSTM

LSTM即长短期记忆网络（long-short term memory networks），其不那么容易发生梯度消失的主要原因在于LSTM内部复杂的“门”(gates)。在计算时，将过程中的梯度进行了抵消。