基于MindSpore的深度学习工具箱开发

**概要方案设计**



苏州同元软控信息技术有限公司

|  |  |  |  |
| --- | --- | --- | --- |
| 编制 |  | 生效日期 |  |
| 审核 |  | 批准 |  |

文件变更摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本** | **变更说明** | **修订** | **审核** | **批准** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 引言 1](#_Toc116397552)

[1.1 编写目的 1](#_Toc116397553)

[1.2 背景介绍 1](#_Toc116397554)

[1.3 参考资料 1](#_Toc116397555)

[2. 产品介绍 1](#_Toc116397556)

[2.1 对标产品 1](#_Toc116397557)

[2.1.1 功能点1 2](#_Toc116397558)

[2.1.2 功能点2 2](#_Toc116397559)

[2.1.3 功能点3 2](#_Toc116397560)

[2.2 应用场景 2](#_Toc116397561)

[3. 产品范围 2](#_Toc116397562)

[3.1 产品目标 3](#_Toc116397563)

[3.2 产品亮点 3](#_Toc116397564)

[4. 系统设计 3](#_Toc116397565)

[4.1 外部接口 3](#_Toc116397566)

[4.1.1 基本构成单元 3](#_Toc116397567)

[4.1.2 容器 4](#_Toc116397568)

[4.1.3 封装层 4](#_Toc116397569)

[4.1.4 卷积神经网络层 5](#_Toc116397570)

[4.1.5 循环神经网络层 5](#_Toc116397571)

[4.1.6 嵌入层 6](#_Toc116397572)

[4.1.7 非线性激活函数层 6](#_Toc116397573)

[4.1.8 线性层 7](#_Toc116397574)

[4.1.9 Droupout层 8](#_Toc116397575)

[4.1.10 归一化层 8](#_Toc116397576)

[4.1.11 池化层 9](#_Toc116397577)

[4.1.12 填充层 9](#_Toc116397578)

[4.1.13 损失函数 10](#_Toc116397579)

[4.1.14 优化器 11](#_Toc116397580)

[4.1.15 评价指标 12](#_Toc116397581)

[4.1.16 动态学习率 14](#_Toc116397582)

[4.1.17 稀疏层 14](#_Toc116397583)

[4.1.18 图像处理层 15](#_Toc116397584)

[4.1.19 矩阵处理 15](#_Toc116397585)

[4.1.20 工具 15](#_Toc116397586)

[4.1.21 数学运算 16](#_Toc116397587)

[4.1.22 梯度 16](#_Toc116397588)

[4.2 内部接口 17](#_Toc116397589)

[4.3 运行环境 17](#_Toc116397590)

[5. 总体研究方案 18](#_Toc116397591)

[5.1 产品总体方案 18](#_Toc116397592)

[5.2 产品基本架构 19](#_Toc116397593)

[5.3 产品研发路线图 20](#_Toc116397594)

[6. 技术路线 20](#_Toc116397595)

# 引言

## 编写目的

本文件描述了项目的总体设计，用于在工具箱开发的过程中对整体结构有详细的描述。

本文件的预期读者为工具箱相关的开发人员以及测试人员。

## 背景介绍

Syslab深度学习工具箱旨在基于华为MindSpore框架，仿照Matlab深度学习工具箱基本结构进行开发，开发和Matlab工具箱中258个函数相同或相似功能的函数。

## 参考资料

* Mindspore官方文档
* MatLab深度学习工具箱帮助文档

# 产品介绍

## 对标产品

Syslab深度学习工具箱对标MATLAB深度学习工具箱，提供了一个用于通过算法、预训练模型和 Api 来设计和实现深度神经网络的框架。可以使用卷积神经网络（ConvNet、CNN）和长短期记忆 (LSTM) 网络对图像、时间序列和文本数据执行分类和回归。您可以使用自动微分、自定义训练循环和共享权重来构建网络架构，如生成对抗网络 (GAN) 和孪生网络。

### 功能点1

自定义神经网络搭建。可以通过定义网络架构并从头开始训练网络，来创建新的用于图像分类和回归任务的深度网络。

### 功能点2

使用预训练网络进行迁移学习。可以使用迁移学习以利用预训练网络所提供的知识来学习新数据中的新模式。

### 功能点3

数据预处理。可以调整图像输入的大小以匹配图像输入层的大小。还可以对数据进行预处理，以增强所需的特征或减少可能导致网络偏差的伪影。例如，对输入数据进行归一化或去噪。

## 应用场景

Syslab深度学习工具箱应用场景包括但不限于：

* 复杂装备工程建模领域：Syslab深度学习工具箱除了可以应用MindSpore已有的案例外，还可以结合同元Sysplorer产品进行联合仿真建模，并采用机理数据融合实现对多构件、复杂系统进行建模，可用于系统健康管理、故障预测、故障诊断等。
* 图像领域：图像识别、图像分割、口罩人脸识别模型、MNIST手写数字识别、哈尔滨工业大学手写汉字训练和识别系统等。
* 自然语言处理：情感识别、鹏城实验室盘古α中文预训练大模型、AI诗人等

# 产品范围

本产品对标MATLAB深度学习工具箱的深度学习神经网络模型实现功能。

## 产品目标

1. 面向Syslab平台的深度学习工具箱，为了使用户在Syslab中能够方便快捷的构建出深度学习模型，并且保证模型精度不低于目前的常用深度学习库。
2. 尽可能达到国内行业标杆的水准。

## 产品亮点

本产品以MWORKS中的工具箱的形式存在，功能包括但不限于：

* 支持MWORKS.Syslab深度学习网络神经模型的自定义搭建。而且操作简单，通过简单的操作就能构建神经网络。
* 支持MWORKS.Syslab深度学习神经网络模型的训练、调参优化以及导入、导出。
* 生态闭环，支持多领域，多工具箱的互操作。能够将MWORKS其他工具箱结合使用，完成例如复杂系统联合仿真建模、故障预测、故障诊断等任务。
* 代码量少，只需短短几行代码，就能在工作中应用深度学习方法，很大程度减少使用者的工作量。
* 支持众多工具和App的应用，例如可视化工具、模型降阶工具，神经网络拟合app等。

# 系统设计

## 外部接口

Syslab深度学习工具箱在开发工程中，将主要依赖mindspore.nn模块，即神经网络Cell，它用于构建神经网络中的预定义构建块或计算单元。

### 基本构成单元

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Cell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Cell.html#mindspore.nn.Cell) | MindSpore中神经网络的基本构成单元。 |
| [mindspore.nn.GraphCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GraphCell.html#mindspore.nn.GraphCell) | 运行从MindIR加载的计算图。 |
| [mindspore.nn.LossBase](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LossBase.html#mindspore.nn.LossBase) | 损失函数的基类。 |
| [mindspore.nn.Optimizer](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Optimizer.html#mindspore.nn.Optimizer) | 用于参数更新的优化器基类。 |

### 容器

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.CellList](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CellList.html#mindspore.nn.CellList) | 构造Cell列表。 |
| [mindspore.nn.SequentialCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SequentialCell.html#mindspore.nn.SequentialCell) | 构造Cell顺序容器。 |

### 封装层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.DistributedGradReducer](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.DistributedGradReducer.html#mindspore.nn.DistributedGradReducer) | 分布式优化器。 |
| [mindspore.nn.DynamicLossScaleUpdateCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.DynamicLossScaleUpdateCell.html#mindspore.nn.DynamicLossScaleUpdateCell) | 用于动态更新损失缩放系数(loss scale)的神经元。 |
| [mindspore.nn.FixedLossScaleUpdateCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.FixedLossScaleUpdateCell.html#mindspore.nn.FixedLossScaleUpdateCell) | 固定损失缩放系数的神经元。 |
| [mindspore.nn.ForwardValueAndGrad](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ForwardValueAndGrad.html#mindspore.nn.ForwardValueAndGrad) | 训练网络的封装。 |
| [mindspore.nn.GetNextSingleOp](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GetNextSingleOp.html#mindspore.nn.GetNextSingleOp) | 用于获取下一条数据的Cell。 |
| [mindspore.nn.MicroBatchInterleaved](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MicroBatchInterleaved.html#mindspore.nn.MicroBatchInterleaved) | Wrap the network with Batch Size. |
| [mindspore.nn.ParameterUpdate](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ParameterUpdate.html#mindspore.nn.ParameterUpdate) | 更新参数的Cell。 |
| [mindspore.nn.PipelineCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.PipelineCell.html#mindspore.nn.PipelineCell) | 将MiniBatch切分成更细粒度的MicroBatch，用于流水线并行的训练中。 |
| [mindspore.nn.TimeDistributed](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.TimeDistributed.html#mindspore.nn.TimeDistributed) | 时间序列封装层。 |
| [mindspore.nn.TrainOneStepCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.TrainOneStepCell.html#mindspore.nn.TrainOneStepCell) | 训练网络封装类。 |
| [mindspore.nn.TrainOneStepWithLossScaleCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.TrainOneStepWithLossScaleCell.html#mindspore.nn.TrainOneStepWithLossScaleCell) | 使用混合精度功能的训练网络。 |
| [mindspore.nn.WithEvalCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.WithEvalCell.html#mindspore.nn.WithEvalCell) | 封装前向网络和损失函数，返回用于计算评估指标的损失函数值、前向输出和标签。 |
| [mindspore.nn.WithGradCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.WithGradCell.html#mindspore.nn.WithGradCell) | Cell that returns the gradients. |
| [mindspore.nn.WithLossCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.WithLossCell.html#mindspore.nn.WithLossCell) | 包含损失函数的Cell。 |

### 卷积神经网络层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Conv1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv1d.html#mindspore.nn.Conv1d) | 一维卷积层。 |
| [mindspore.nn.Conv1dTranspose](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv1dTranspose.html#mindspore.nn.Conv1dTranspose) | 一维转置卷积层。 |
| [mindspore.nn.Conv2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv2d.html#mindspore.nn.Conv2d) | 二维卷积层。 |
| [mindspore.nn.Conv2dTranspose](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv2dTranspose.html#mindspore.nn.Conv2dTranspose) | 二维转置卷积层。 |
| [mindspore.nn.Conv3d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv3d.html#mindspore.nn.Conv3d) | 三维卷积层。 |
| [mindspore.nn.Conv3dTranspose](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Conv3dTranspose.html#mindspore.nn.Conv3dTranspose) | 三维转置卷积层。 |
| [mindspore.nn.Unfold](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Unfold.html#mindspore.nn.Unfold) | 从图像中提取滑窗的区域块。 |

### 循环神经网络层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.RNN](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RNN.html#mindspore.nn.RNN) | 循环神经网络（RNN）层，其使用的激活函数为tanh或relu。 |
| [mindspore.nn.RNNCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RNNCell.html#mindspore.nn.RNNCell) | 循环神经网络单元，激活函数是tanh或relu。 |
| [mindspore.nn.GRU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GRU.html#mindspore.nn.GRU) | GRU（Gate Recurrent Unit）称为门控循环单元网络，是循环神经网络（Recurrent Neural Network, RNN）的一种。 |
| [mindspore.nn.GRUCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GRUCell.html#mindspore.nn.GRUCell) | GRU（Gate Recurrent Unit）称为门控循环单元。 |
| [mindspore.nn.LSTM](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LSTM.html#mindspore.nn.LSTM) | 长短期记忆（LSTM）网络，根据输出序列和给定的初始状态计算输出序列和最终状态。 |
| [mindspore.nn.LSTMCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LSTMCell.html#mindspore.nn.LSTMCell) | 长短期记忆网络单元（LSTMCell）。 |

### 嵌入层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Embedding](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Embedding.html#mindspore.nn.Embedding) | 嵌入层。 |
| [mindspore.nn.EmbeddingLookup](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.EmbeddingLookup.html#mindspore.nn.EmbeddingLookup) | 嵌入查找层。 |
| [mindspore.nn.MultiFieldEmbeddingLookup](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MultiFieldEmbeddingLookup.html#mindspore.nn.MultiFieldEmbeddingLookup) | 根据指定的索引和字段ID，返回输入Tensor的切片。 |

### 非线性激活函数层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.CELU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CELU.html#mindspore.nn.CELU) | Continuously differentiable exponential linear units activation function. |
| [mindspore.nn.ELU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ELU.html#mindspore.nn.ELU) | 指数线性单元激活函数（Exponential Linear Unit activation function）。 |
| [mindspore.nn.FastGelu](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.FastGelu.html#mindspore.nn.FastGelu) | 快速高斯误差线性单元激活函数（Fast Gaussian Error Linear Units activation function）。 |
| [mindspore.nn.GELU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GELU.html#mindspore.nn.GELU) | 高斯误差线性单元激活函数（Gaussian error linear unit activation function）。 |
| [mindspore.nn.Hardtanh](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Hardtanh.html#mindspore.nn.Hardtanh) | Hardtanh激活函数。 |
| [mindspore.nn.HShrink](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.HShrink.html#mindspore.nn.HShrink) | Hard Shrink激活函数，按输入元素计算输出，公式定义如下：。 |
| [mindspore.nn.HSigmoid](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.HSigmoid.html#mindspore.nn.HSigmoid) | Hard Sigmoid激活函数，按元素计算输出。 |
| [mindspore.nn.HSwish](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.HSwish.html#mindspore.nn.HSwish) | Hard Swish激活函数。 |
| [mindspore.nn.LeakyReLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LeakyReLU.html#mindspore.nn.LeakyReLU) | Leaky ReLU激活函数。 |
| [mindspore.nn.LogSigmoid](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LogSigmoid.html#mindspore.nn.LogSigmoid) | Log Sigmoid激活函数。 |
| [mindspore.nn.LogSoftmax](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LogSoftmax.html#mindspore.nn.LogSoftmax) | Log Softmax激活函数。 |
| [mindspore.nn.LRN](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LRN.html#mindspore.nn.LRN) | 局部响应归一化操作LRN(Local Response Normalization)。 |
| [mindspore.nn.Mish](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Mish.html#mindspore.nn.Mish) | 逐元素计算输入Tensor的MISH（Self Regularized Non-Monotonic Neural Activation Function 自正则化非单调神经激活函数）。 |
| [mindspore.nn.Softsign](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Softsign.html#mindspore.nn.Softsign) | Softsign激活函数。 |
| [mindspore.nn.PReLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.PReLU.html#mindspore.nn.PReLU) | PReLU激活层（PReLU Activation Operator）。 |
| [mindspore.nn.ReLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ReLU.html#mindspore.nn.ReLU) | 修正线性单元激活函数（Rectified Linear Unit activation function）。 |
| [mindspore.nn.ReLU6](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ReLU6.html#mindspore.nn.ReLU6) | ReLU6激活函数。 |
| [mindspore.nn.RReLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RReLU.html#mindspore.nn.RReLU) | Randomized Leaky ReLU激活函数。 |
| [mindspore.nn.SeLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SeLU.html#mindspore.nn.SeLU) | 激活函数selu（Scaled exponential Linear Unit）。 |
| [mindspore.nn.SiLU](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SiLU.html#mindspore.nn.SiLU) | SiLU激活函数。 |
| [mindspore.nn.Sigmoid](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Sigmoid.html#mindspore.nn.Sigmoid) | Sigmoid激活函数。 |
| [mindspore.nn.Softmax](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Softmax.html#mindspore.nn.Softmax) | Softmax函数，它是二分类函数 [mindspore.nn.Sigmoid](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Sigmoid.html#mindspore.nn.Sigmoid) 在多分类上的推广，目的是将多分类的结果以概率的形式展现出来。 |
| [mindspore.nn.SoftShrink](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SoftShrink.html#mindspore.nn.SoftShrink) | SoftShrink激活函数。 |
| [mindspore.nn.Tanh](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Tanh.html#mindspore.nn.Tanh) | Tanh激活函数。 |
| [mindspore.nn.Tanhshrink](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Tanhshrink.html#mindspore.nn.Tanhshrink) | Tanhshrink激活函数。 |
| [mindspore.nn.Threshold](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Threshold.html#mindspore.nn.Threshold) | Threshold激活函数，按元素计算输出。 |

### 线性层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Dense](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Dense.html#mindspore.nn.Dense) | 全连接层。 |
| [mindspore.nn.BiDense](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BiDense.html#mindspore.nn.BiDense) | 双线性全连接层。 |

### Droupout层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Dropout](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Dropout.html#mindspore.nn.Dropout) | 随机丢弃层。 |
| [mindspore.nn.Dropout2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Dropout2d.html#mindspore.nn.Dropout2d) | 在训练期间，以服从伯努利分布的概率 p 随机将输入Tensor的某些通道归零（对于形状为 NCHW 的四维Tensor，其通道特征图指的是后两维 HW 形状的二维特征图）。 |
| [mindspore.nn.Dropout3d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Dropout3d.html#mindspore.nn.Dropout3d) | 在训练期间，以服从伯努利分布的概率 p 随机将输入Tensor的某些通道归零（对于形状为 NCDHW 的 5D Tensor。 |

### 归一化层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.BatchNorm1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BatchNorm1d.html#mindspore.nn.BatchNorm1d) | 对输入的二维数据进行批归一化(Batch Normalization Layer)。 |
| [mindspore.nn.BatchNorm2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BatchNorm2d.html#mindspore.nn.BatchNorm2d) | 对输入的四维数据进行批归一化(Batch Normalization Layer)。 |
| [mindspore.nn.BatchNorm3d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BatchNorm3d.html#mindspore.nn.BatchNorm3d) | 对输入的五维数据进行批归一化(Batch Normalization Layer)。 |
| [mindspore.nn.GroupNorm](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.GroupNorm.html#mindspore.nn.GroupNorm) | 在mini-batch输入上进行组归一化。 |
| [mindspore.nn.InstanceNorm1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.InstanceNorm1d.html#mindspore.nn.InstanceNorm1d) | 对三维输入实现实例归一化（Instance Normalization Layer）。 |
| [mindspore.nn.InstanceNorm2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.InstanceNorm2d.html#mindspore.nn.InstanceNorm2d) | 对四维输入实现实例归一化（Instance Normalization Layer）。 |
| [mindspore.nn.InstanceNorm3d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.InstanceNorm3d.html#mindspore.nn.InstanceNorm3d) | 对五维输入实现实例归一化（Instance Normalization Layer）。 |
| [mindspore.nn.LayerNorm](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LayerNorm.html#mindspore.nn.LayerNorm) | 在mini-batch输入上应用层归一化（Layer Normalization）。 |
| [mindspore.nn.SyncBatchNorm](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SyncBatchNorm.html#mindspore.nn.SyncBatchNorm) | 在N维输入上进行跨设备同步批归一化（Batch Normalization，BN）。 |

### 池化层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.AdaptiveAvgPool1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaptiveAvgPool1d.html#mindspore.nn.AdaptiveAvgPool1d) | 对输入的多维数据进行一维平面上的自适应平均池化运算。 |
| [mindspore.nn.AdaptiveAvgPool2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaptiveAvgPool2d.html#mindspore.nn.AdaptiveAvgPool2d) | 2维自适应平均池化。 |
| [mindspore.nn.AdaptiveMaxPool1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaptiveMaxPool1d.html#mindspore.nn.AdaptiveMaxPool1d) | 对输入的多维数据进行一维平面上的自适应最大池化运算。 |
| [mindspore.nn.AdaptiveMaxPool2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaptiveMaxPool2d.html#mindspore.nn.AdaptiveMaxPool2d) | 二维自适应最大池化运算。 |
| [mindspore.nn.AvgPool1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AvgPool1d.html#mindspore.nn.AvgPool1d) | 对输入的多维数据进行一维平面上的平均池化运算。 |
| [mindspore.nn.AvgPool2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AvgPool2d.html#mindspore.nn.AvgPool2d) | 对输入的多维数据进行二维的平均池化运算。 |
| [mindspore.nn.MaxPool1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MaxPool1d.html#mindspore.nn.MaxPool1d) | 对输入的多维数据进行一维平面上的最大池化运算。 |
| [mindspore.nn.MaxPool2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MaxPool2d.html#mindspore.nn.MaxPool2d) | 对输入的多维数据进行二维的最大池化运算。 |

### 填充层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Pad](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Pad.html#mindspore.nn.Pad) | 根据 paddings 和 mode 对输入进行填充。 |
| [mindspore.nn.ConstantPad1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ConstantPad1d.html#mindspore.nn.ConstantPad1d) | 将给定的常量填充到多维输入数据的最后一维。 |
| [mindspore.nn.ConstantPad2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ConstantPad2d.html#mindspore.nn.ConstantPad2d) | 将给定的常量填充到多维输入数据的最后两维。 |
| [mindspore.nn.ConstantPad3d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ConstantPad3d.html#mindspore.nn.ConstantPad3d) | 将给定的常量填充到多维输入数据的最后三维。 |
| [mindspore.nn.ReflectionPad1d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ReflectionPad1d.html#mindspore.nn.ReflectionPad1d) | 根据 padding 对输入 x 进行填充。 |
| [mindspore.nn.ReflectionPad2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ReflectionPad2d.html#mindspore.nn.ReflectionPad2d) | 根据 padding 对输入 x 进行填充。 |
| [mindspore.nn.ZeroPad2d](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ZeroPad2d.html#mindspore.nn.ZeroPad2d) | 将零填充到多维输入数据的最后两维。 |

### 损失函数

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.BCELoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BCELoss.html#mindspore.nn.BCELoss) | 计算目标值和预测值之间的二值交叉熵损失值。 |
| [mindspore.nn.BCEWithLogitsLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BCEWithLogitsLoss.html#mindspore.nn.BCEWithLogitsLoss) | 输入经过sigmoid激活函数后作为预测值，BCEWithLogitsLoss 计算预测值和目标值之间的二值交叉熵损失。 |
| [mindspore.nn.CosineEmbeddingLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CosineEmbeddingLoss.html#mindspore.nn.CosineEmbeddingLoss) | 余弦相似度损失函数，用于测量两个Tensor之间的相似性。 |
| [mindspore.nn.CrossEntropyLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CrossEntropyLoss.html#mindspore.nn.CrossEntropyLoss) | 计算预测值和目标值之间的交叉熵损失。 |
| [mindspore.nn.DiceLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.DiceLoss.html#mindspore.nn.DiceLoss) | Dice系数是一个集合相似性loss，用于计算两个样本之间的相似性。 |
| [mindspore.nn.FocalLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.FocalLoss.html#mindspore.nn.FocalLoss) | FocalLoss函数。 |
| [mindspore.nn.HuberLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.HuberLoss.html#mindspore.nn.HuberLoss) | HuberLoss计算预测值和目标值之间的误差。 |
| [mindspore.nn.L1Loss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.L1Loss.html#mindspore.nn.L1Loss) | L1Loss用于计算预测值和目标值之间的平均绝对误差。 |
| [mindspore.nn.MSELoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MSELoss.html#mindspore.nn.MSELoss) | 用于计算预测值与标签值之间的均方误差。 |
| [mindspore.nn.MultiClassDiceLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MultiClassDiceLoss.html#mindspore.nn.MultiClassDiceLoss) | 对于多标签问题，可以将标签通过one-hot编码转换为多个二分类标签。 |
| [mindspore.nn.NLLLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.NLLLoss.html#mindspore.nn.NLLLoss) | 计算预测值和目标值之间的负对数似然损失。 |
| [mindspore.nn.RMSELoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RMSELoss.html#mindspore.nn.RMSELoss) | RMSELoss用来测量 xx 和 yy 元素之间的均方根误差，其中 xx 是输入Tensor， yy 是目标值。 |
| [mindspore.nn.SampledSoftmaxLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SampledSoftmaxLoss.html#mindspore.nn.SampledSoftmaxLoss) | 抽样交叉熵损失函数。 |
| [mindspore.nn.SmoothL1Loss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SmoothL1Loss.html#mindspore.nn.SmoothL1Loss) | SmoothL1损失函数，如果预测值和目标值的逐个元素绝对误差小于设定阈值 beta 则用平方项，否则用绝对误差项。 |
| [mindspore.nn.SoftMarginLoss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SoftMarginLoss.html#mindspore.nn.SoftMarginLoss) | 针对二分类问题的损失函数。 |
| [mindspore.nn.SoftmaxCrossEntropyWithLogits](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SoftmaxCrossEntropyWithLogits.html#mindspore.nn.SoftmaxCrossEntropyWithLogits) | 计算预测值与真实值之间的交叉熵。 |

### 优化器

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Adadelta](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Adadelta.html#mindspore.nn.Adadelta) | Adadelta算法的实现。 |
| [mindspore.nn.Adagrad](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Adagrad.html#mindspore.nn.Adagrad) | Adagrad算法的实现。 |
| [mindspore.nn.Adam](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Adam.html#mindspore.nn.Adam) | Adaptive Moment Estimation (Adam)算法的实现。 |
| [mindspore.nn.AdaMax](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaMax.html#mindspore.nn.AdaMax) | AdaMax算法是基于无穷范数的Adam的一种变体。 |
| [mindspore.nn.AdamOffload](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdamOffload.html#mindspore.nn.AdamOffload) | 此优化器在主机CPU上运行Adam优化算法，设备上仅执行网络参数的更新，最大限度地降低内存成本。 |
| [mindspore.nn.AdamWeightDecay](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdamWeightDecay.html#mindspore.nn.AdamWeightDecay) | 权重衰减Adam算法的实现。 |
| [mindspore.nn.AdaSumByDeltaWeight](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaSumByDeltaWeightWrapCell.html#mindspore.nn.AdaSumByDeltaWeightWrapCell)  WrapCell | Adaptive Summation (AdaSum)算法的实现，根据更新前后的参数差计算。 |
| [mindspore.nn.AdaSumByGradWrapCell](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.AdaSumByGradWrapCell.html#mindspore.nn.AdaSumByGradWrapCell) | Adaptive Summation (AdaSum)算法的实现，根据梯度计算。 |
| [mindspore.nn.ASGD](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ASGD.html#mindspore.nn.ASGD) | Implements Average Stochastic Gradient Descent. |
| [mindspore.nn.FTRL](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.FTRL.html#mindspore.nn.FTRL) | FTRL算法实现。 |
| [mindspore.nn.Lamb](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Lamb.html#mindspore.nn.Lamb) | LAMB（Layer-wise Adaptive Moments optimizer for Batching training，用于批训练的分层自适应矩优化器）算法的实现。 |
| [mindspore.nn.LARS](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LARS.html#mindspore.nn.LARS) | LARS算法的实现。 |
| [mindspore.nn.LazyAdam](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.LazyAdam.html#mindspore.nn.LazyAdam) | Adaptive Moment Estimation (Adam)算法的实现。 |
| [mindspore.nn.Momentum](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Momentum.html#mindspore.nn.Momentum) | Momentum算法的实现。 |
| [mindspore.nn.ProximalAdagrad](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ProximalAdagrad.html#mindspore.nn.ProximalAdagrad) | ProximalAdagrad算法的实现。 |
| [mindspore.nn.RMSProp](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RMSProp.html#mindspore.nn.RMSProp) | 均方根传播（RMSProp）算法的实现。 |
| [mindspore.nn.Rprop](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Rprop.html#mindspore.nn.Rprop) | Implements Resilient backpropagation. |
| [mindspore.nn.SGD](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SGD.html#mindspore.nn.SGD) | 随机梯度下降的实现。 |
| [mindspore.nn.thor](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.thor.html#mindspore.nn.thor) | 通过二阶算法THOR更新参数。 |

### 评价指标

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Accuracy](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Accuracy.html#mindspore.nn.Accuracy) | 计算数据分类的正确率，包括二分类和多分类。 |
| [mindspore.nn.auc](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.auc.html#mindspore.nn.auc) | 使用梯形法则计算曲线下面积AUC（Area Under the Curve，AUC）。 |
| [mindspore.nn.BleuScore](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.BleuScore.html#mindspore.nn.BleuScore) | 计算具有一个或多个引用的机器翻译文本的BLEU分数。 |
| [mindspore.nn.ConfusionMatrix](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ConfusionMatrix.html#mindspore.nn.ConfusionMatrix) | 计算混淆矩阵(confusion matrix)，通常用于评估分类模型的性能，包括二分类和多分类场景。 |
| [mindspore.nn.ConfusionMatrixMetric](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ConfusionMatrixMetric.html#mindspore.nn.ConfusionMatrixMetric) | 计算与混淆矩阵相关的度量。 |
| [mindspore.nn.CosineSimilarity](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CosineSimilarity.html#mindspore.nn.CosineSimilarity) | 计算余弦相似度。 |
| [mindspore.nn.Dice](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Dice.html#mindspore.nn.Dice) | 集合相似性度量。 |
| [mindspore.nn.F1](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.F1.html#mindspore.nn.F1) | 计算F1 score。 |
| [mindspore.nn.Fbeta](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Fbeta.html#mindspore.nn.Fbeta) | 计算Fbeta评分。 |
| [mindspore.nn.HausdorffDistance](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.HausdorffDistance.html#mindspore.nn.HausdorffDistance) | 计算Hausdorff距离。 |
| [mindspore.nn.get\_metric\_fn](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.get_metric_fn.html#mindspore.nn.get_metric_fn) | 根据输入的 name 获取metric的方法。 |
| [mindspore.nn.Loss](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Loss.html#mindspore.nn.Loss) | 计算loss的平均值。 |
| [mindspore.nn.MAE](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MAE.html#mindspore.nn.MAE) | 计算平均绝对误差MAE（Mean Absolute Error）。 |
| [mindspore.nn.MeanSurfaceDistance](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MeanSurfaceDistance.html#mindspore.nn.MeanSurfaceDistance) | 计算从 y\_pred 到 y 的平均表面距离。 |
| [mindspore.nn.Metric](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Metric.html#mindspore.nn.Metric) | 用于计算评估指标的基类。 |
| [mindspore.nn.MSE](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MSE.html#mindspore.nn.MSE) | 测量均方差MSE（Mean Squared Error）。 |
| [mindspore.nn.names](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.names.html#mindspore.nn.names) | 获取所有metric的名称。 |
| [mindspore.nn.OcclusionSensitivity](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.OcclusionSensitivity.html#mindspore.nn.OcclusionSensitivity) | 用于计算神经网络对给定图像的遮挡灵敏度（Occlusion Sensitivity），表示了图像的哪些部分对神经网络的分类决策最重要。 |
| [mindspore.nn.Perplexity](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Perplexity.html#mindspore.nn.Perplexity) | 计算困惑度（perplexity）。 |
| [mindspore.nn.Precision](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Precision.html#mindspore.nn.Precision) | 计算数据分类的精度，包括单标签场景和多标签场景。 |
| [mindspore.nn.Recall](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Recall.html#mindspore.nn.Recall) | 计算数据分类的召回率，包括单标签场景和多标签场景。 |
| [mindspore.nn.ROC](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ROC.html#mindspore.nn.ROC) | 计算ROC曲线。 |
| [mindspore.nn.RootMeanSquareDistance](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.RootMeanSquareDistance.html#mindspore.nn.RootMeanSquareDistance) | 计算从 y\_pred 到 y 的均方根表面距离。 |
| [mindspore.nn.rearrange\_inputs](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.rearrange_inputs.html#mindspore.nn.rearrange_inputs) | 此装饰器用于根据类的 indexes 属性对输入重新排列。 |
| [mindspore.nn.Top1CategoricalAccuracy](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Top1CategoricalAccuracy.html#mindspore.nn.Top1CategoricalAccuracy) | 计算top-1分类正确率。 |
| [mindspore.nn.Top5CategoricalAccuracy](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Top5CategoricalAccuracy.html#mindspore.nn.Top5CategoricalAccuracy) | 计算top-5分类正确率。 |
| [mindspore.nn.TopKCategoricalAccuracy](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.TopKCategoricalAccuracy.html#mindspore.nn.TopKCategoricalAccuracy) | 计算top-k分类正确率。 |

### 动态学习率

#### LearningRateSchedule类

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.CosineDecayLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CosineDecayLR.html#mindspore.nn.CosineDecayLR) | 基于余弦衰减函数计算学习率。 |
| [mindspore.nn.ExponentialDecayLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ExponentialDecayLR.html#mindspore.nn.ExponentialDecayLR) | 基于指数衰减函数计算学习率。 |
| [mindspore.nn.InverseDecayLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.InverseDecayLR.html#mindspore.nn.InverseDecayLR) | 基于逆时衰减函数计算学习率。 |
| [mindspore.nn.NaturalExpDecayLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.NaturalExpDecayLR.html#mindspore.nn.NaturalExpDecayLR) | 基于自然指数衰减函数计算学习率。 |
| [mindspore.nn.PolynomialDecayLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.PolynomialDecayLR.html#mindspore.nn.PolynomialDecayLR) | 基于多项式衰减函数计算学习率。 |
| [mindspore.nn.WarmUpLR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.WarmUpLR.html#mindspore.nn.WarmUpLR) | 预热学习率。 |

#### Dynamic LR函数

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.cosine\_decay\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.cosine_decay_lr.html#mindspore.nn.cosine_decay_lr) | 基于余弦衰减函数计算学习率。 |
| [mindspore.nn.exponential\_decay\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.exponential_decay_lr.html#mindspore.nn.exponential_decay_lr) | 基于指数衰减函数计算学习率。 |
| [mindspore.nn.inverse\_decay\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.inverse_decay_lr.html#mindspore.nn.inverse_decay_lr) | 基于逆时衰减函数计算学习率。 |
| [mindspore.nn.natural\_exp\_decay\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.natural_exp_decay_lr.html#mindspore.nn.natural_exp_decay_lr) | 基于自然指数衰减函数计算学习率。 |
| [mindspore.nn.piecewise\_constant\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.piecewise_constant_lr.html#mindspore.nn.piecewise_constant_lr) | 获取分段常量学习率。 |
| [mindspore.nn.polynomial\_decay\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.polynomial_decay_lr.html#mindspore.nn.polynomial_decay_lr) | 基于多项式衰减函数计算学习率。 |
| [mindspore.nn.warmup\_lr](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.warmup_lr.html#mindspore.nn.warmup_lr) | 预热学习率。 |

### 稀疏层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.SparseTensorDenseMatmul](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SparseTensorDenseMatmul.html#mindspore.nn.SparseTensorDenseMatmul) | 稀疏矩阵 a 乘以稠密矩阵 b 。 |
| [mindspore.nn.SparseToDense](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SparseToDense.html#mindspore.nn.SparseToDense) | 将稀疏Tensor(COO Tensor)转换为稠密Tensor。 |

### 图像处理层

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.CentralCrop](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.CentralCrop.html#mindspore.nn.CentralCrop) | 根据指定比例裁剪出图像的中心区域。 |
| [mindspore.nn.ImageGradients](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ImageGradients.html#mindspore.nn.ImageGradients) | 计算每个颜色通道的图像渐变，返回为两个Tensor，分别表示高和宽方向上的变化率。 |
| [mindspore.nn.MSSSIM](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MSSSIM.html#mindspore.nn.MSSSIM) | 多尺度计算两个图像之间的结构相似性（SSIM）。 |
| [mindspore.nn.PSNR](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.PSNR.html#mindspore.nn.PSNR) | 在批处理中计算两个图像的峰值信噪比（PSNR）。 |
| [mindspore.nn.ResizeBilinear](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ResizeBilinear.html#mindspore.nn.ResizeBilinear) | 使用双线性插值调整输入Tensor为指定的大小。 |
| [mindspore.nn.SSIM](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.SSIM.html#mindspore.nn.SSIM) | 计算两个图像之间的结构相似性（SSIM）。 |

### 矩阵处理

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.MatrixDiag](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MatrixDiag.html#mindspore.nn.MatrixDiag) | 根据对角线值返回一批对角矩阵。 |
| [mindspore.nn.MatrixDiagPart](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MatrixDiagPart.html#mindspore.nn.MatrixDiagPart) | 返回批对角矩阵的对角线部分。 |
| [mindspore.nn.MatrixSetDiag](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.MatrixSetDiag.html#mindspore.nn.MatrixSetDiag) | 将输入的对角矩阵的对角线值置换为输入的对角线值。 |

### 工具

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.ClipByNorm](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ClipByNorm.html#mindspore.nn.ClipByNorm) | 对输入Tensor的值进行裁剪，使用 L2L2 范数控制梯度。 |
| [mindspore.nn.Flatten](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Flatten.html#mindspore.nn.Flatten) | 对输入Tensor的第0维之外的维度进行展平操作。 |
| [mindspore.nn.get\_activation](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.get_activation.html#mindspore.nn.get_activation) | 获取激活函数。 |
| [mindspore.nn.L1Regularizer](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.L1Regularizer.html#mindspore.nn.L1Regularizer) | 对权重计算L1正则化。 |
| [mindspore.nn.Norm](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Norm.html#mindspore.nn.Norm) | 计算向量的范数，目前包括欧几里得范数，即 L2L2-norm。 |
| [mindspore.nn.OneHot](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.OneHot.html#mindspore.nn.OneHot) | 对输入进行one-hot编码并返回。 |
| [mindspore.nn.Range](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Range.html#mindspore.nn.Range) | 根据指定步长在范围[start, limit)中创建数字序列。 |
| [mindspore.nn.Roll](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Roll.html#mindspore.nn.Roll) | 沿轴移动Tensor的元素。 |
| [mindspore.nn.Tril](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Tril.html#mindspore.nn.Tril) | 返回一个Tensor，指定主对角线以上的元素被置为零。 |
| [mindspore.nn.Triu](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Triu.html#mindspore.nn.Triu) | 返回一个Tensor，指定主对角线以下的元素被置为0。 |

### 数学运算

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Moments](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Moments.html#mindspore.nn.Moments) | 沿指定轴 axis 计算输入 x 的均值和方差。 |
| [mindspore.nn.ReduceLogSumExp](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.ReduceLogSumExp.html#mindspore.nn.ReduceLogSumExp) | Reduces a dimension of a tensor by calculating exponential for all elements in the dimension, then calculate logarithm of the sum. |

### 梯度

|  |  |
| --- | --- |
| 接口名 | 概述 |
| [mindspore.nn.Jvp](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Jvp.html#mindspore.nn.Jvp) | 计算给定网络的雅可比向量积(Jacobian-vector product, JVP)。 |
| [mindspore.nn.Vjp](https://www.mindspore.cn/docs/zh-CN/r1.8/api_python/nn/mindspore.nn.Vjp.html#mindspore.nn.Vjp) | 计算给定网络的向量雅可比积(vector-Jacobian product, VJP)。 |

## 内部接口

|  |  |
| --- | --- |
| 接口名 | 概述 |
| TyPlot | Syslab平台可视化工具 |
| TyStatistics | Syalab平台数学统计库 |

## 运行环境

Julia：v1.7.3

|  |  |  |
| --- | --- | --- |
| 依赖库 | 版本 | 概述 |
| TyPlot | 0.7.0 | Julia可视化 |
| PyCall | 1.93.1 | Julia调取python接口 |
| TyStatistics | 0.1.0 | 数学统计库 |

Python：v3.9.7

|  |  |  |
| --- | --- | --- |
| 依赖库 | 版本 | 概述 |
| MindSpore | 1.8.1 | 神经网络框架 |
| MindSpore-hub | 1.6.0 | MindSpore预训练模型框架 |
| Networkx | 2.8.6 | 神经网络图绘制 |
| MiniSom | 2.3.0 | 自组织地图构建 |

# 总体研究方案

## 产品总体方案



* 功能原型：Syslab深度学习工具箱依赖MindSpore框架开发，将mindspore.nn和Syslab科学计算平台作为工具箱的开发功能原型，在开发的同时对标Matlab深度学习工具箱，以期达到性能比肩及超越。
* 应用流程：Syslab深度学习工具箱采用同MindSpore相同的应用流程，从数据层的数据采集到模型层的模型构建，再到训练层的模型训练，最后通过可视化层和应用层将模型可视化和部署。同时，在建模过程中，可采用Sysplorer和Syslab其他工具箱作为依赖完善整个流程。
* 应用场景：Syslab深度学习工具箱除了可以应用MindSpore已有的案例外，还可以结合同元Sysplorer产品进行联合仿真建模，并采用机理数据融合实现对多构件、复杂系统进行建模，可用于系统健康管理、故障预测、故障诊断等。

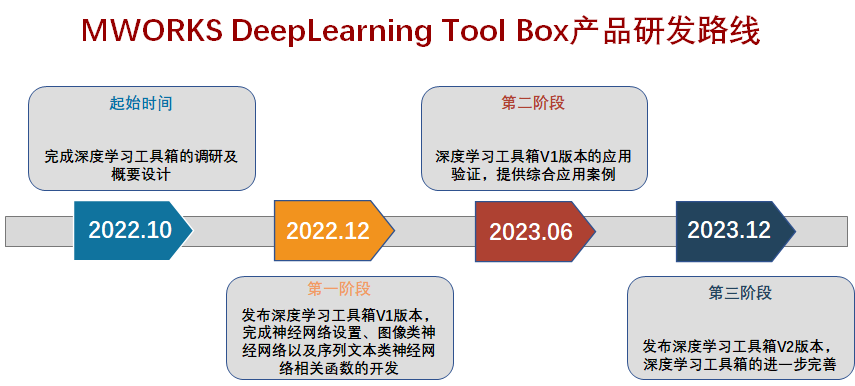
## 产品基本架构



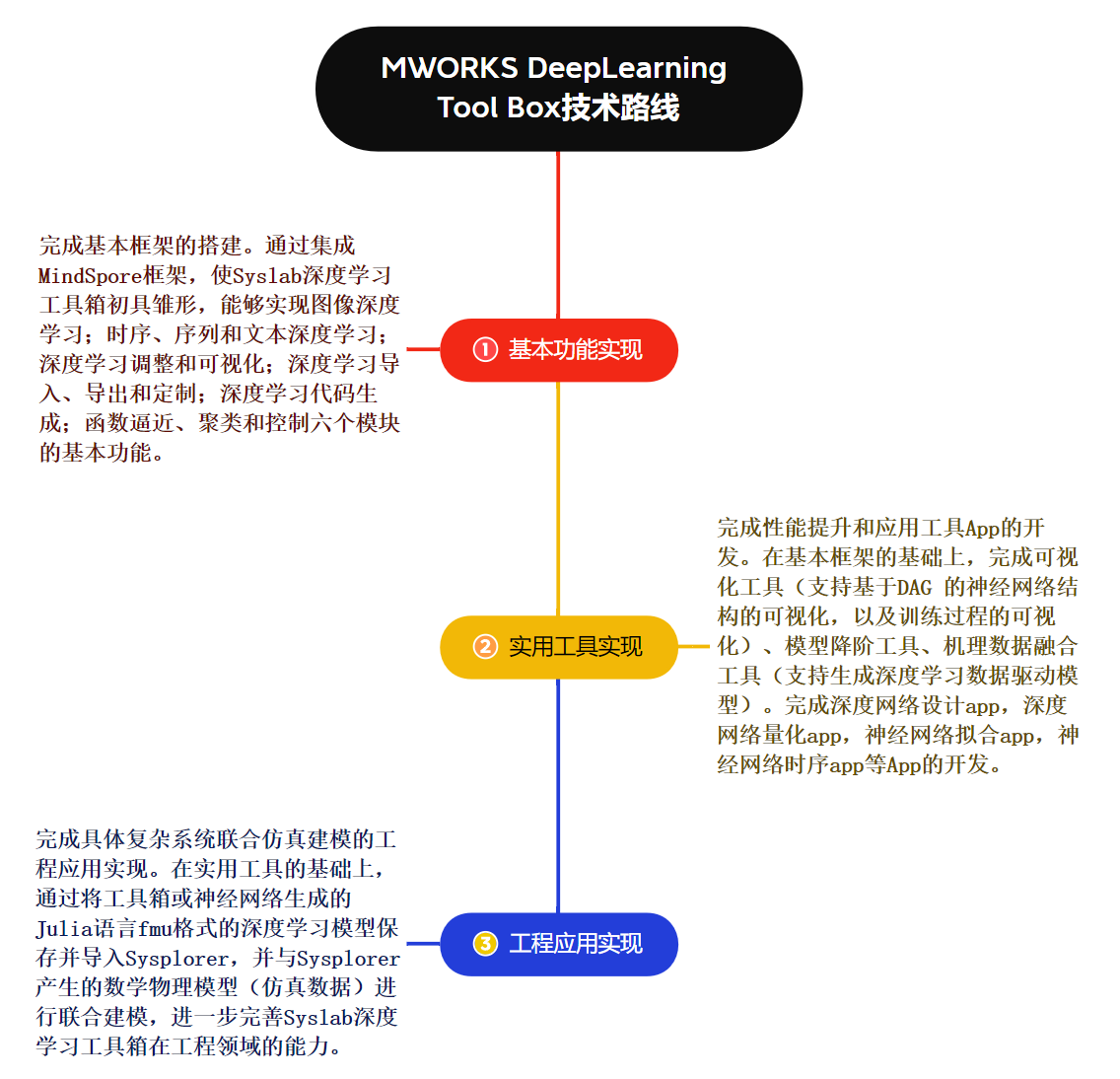
* AI芯片：支持通用并行计算架构，支持CPU+AI芯片的加速平台
* 数据层：数据结构的定义，数据存储的管理
* 模型层：支持各类神经网络模型的构建，包含各种网络层和激活函数
* 训练层：支持训练参数的设置，以及神经网络权重参数的优化计算
* 可视化：支持基于DAG 的神经网络结构的可视化，以及训练过程的可视化
* 应用app：深度网络设计app，深度网络量化app，神经网络拟合app，神经网络时序app。
* 导入导出接口：支持与 TensorFlow™ 和 PyTorch 交换模型，并从 TensorFlow-Keras 和 Caffe 导入模型
* 代码生成：自动生成 C/C++ 代码用于嵌入式部署

## 产品研发路线图

Syslab 深度学习工具箱产品路线图将深度学习工具箱研发分为三个阶段，第一阶段基于华为MindSpore库，仿照Matlab深度学习工具箱相关的功能函数进行开发，完成神经网络设置、图像类神经网络以及序列文本类神经网络相关函数开发。第二阶段主要完成深度学习工具箱V1版本应用验证，并完成综合应用案例的开发。第三阶段，对已完成的V1版本进行代码优化，依据客户使用反馈进行改进，提高算法的运行速度与功能的易用性，发布深度学习工具箱V2版本。



# 技术路线



Syslab 深度学习工具箱将分为三个阶段进行技术更新与能力提升：

第一阶段：基本功能实现

完成基本框架的搭建。通过集成MindSpore框架，使Syslab深度学习工具箱初具雏形，能够实现图像深度学习；时序、序列和文本深度学习；深度学习调整和可视化；深度学习导入、导出和定制；深度学习代码生成；函数逼近、聚类和控制六个模块的基本功能。

第二阶段：实用工具实现

完成性能提升和应用工具App的开发。在基本框架的基础上，完成可视化工具（支持基于DAG 的神经网络结构的可视化，以及训练过程的可视化）、模型降阶工具、机理数据融合工具（支持生成深度学习数据驱动模型）。完成深度网络设计app，深度网络量化app，神经网络拟合app，神经网络时序app等App的开发。

第三阶段：工程应用实现

完成具体复杂系统联合仿真建模的工程应用实现。在实用工具的基础上，通过将工具箱或神经网络生成的Julia语言fmu格式的深度学习模型保存并导入Sysplorer，并与Sysplorer产生的数学物理模型（仿真数据）进行联合建模，进一步完善Syslab深度学习工具箱在工程领域的能力。