



# MoEx-工具复现

《On Feature Normalization and Data Augmentation》

汇报人：陈家志191250009

# 目录大纲

CONTENT

1

核心算法介绍

2

功能模块介绍

3

输入输出理解

4

ELSE



01

# 核心算法介绍



## 核心算法思想

- 除了特征数据本身以外，正则化（归一化）过程中产生的一阶矩（ $\mu$ ）、二阶矩（ $\sigma$ ）同样捕获了图像样本的深层特征，显露了他们对应的分类标签。MoEx企图运用这种内在蕴含，来进行数据增扩增加模型训练的鲁棒性

为什么要正则化？

- 归一后的特征数据拥有 均值0，标准差1的良好数学特点，意味着对应数据可以同时使用实例内(intra-instance)的正则化以及实例间的正则化(inter-instance)

## 核心算法原理概述

1. 定义函数 $F$ ，以模型中第  $\ell$  层的特征作为输入，以归一化的特征数据、一阶矩、二阶矩为输出
2. 定义函数 $F^{-1}$ 作为 $F$ 的逆函数

$$(\hat{\mathbf{h}}_i^\ell, \boldsymbol{\mu}_i^\ell, \boldsymbol{\sigma}_i^\ell) = F(\mathbf{h}_i^\ell), \quad \mathbf{h}_i^\ell = F^{-1}(\hat{\mathbf{h}}_i^\ell, \boldsymbol{\mu}_i^\ell, \boldsymbol{\sigma}_i^\ell).$$

3. 那么对任意样本 $A, B$ , 我们都可以通过以下的公式来把 $B$ 的矩特征插入到 $A$ 的特征数据中

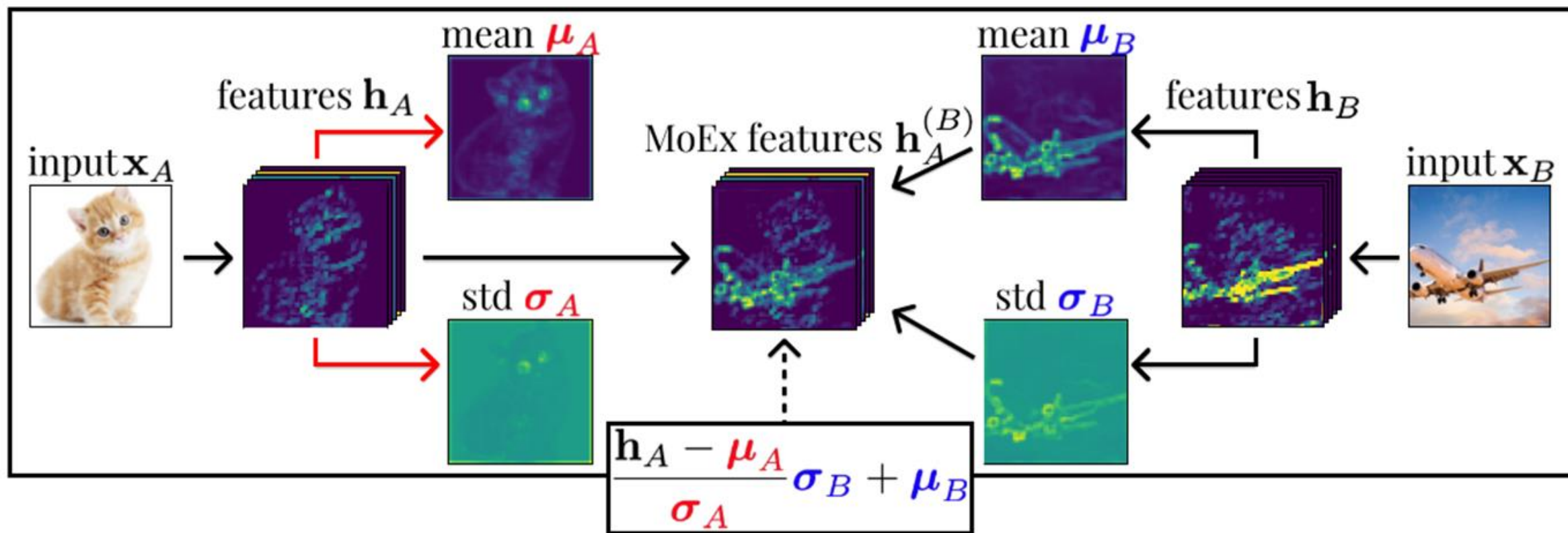
$$\mathbf{x}_A: \mathbf{h}_A^{(B)} = F^{-1}(\hat{\mathbf{h}}_A, \boldsymbol{\mu}_B, \boldsymbol{\sigma}_B).$$

4. 相应的损失函数  $\ell$  应该对这种更改有一定程度的反馈，这个程度我们用参数  $\lambda$  来人为界定

$$\lambda \cdot \ell(\mathbf{h}_A^{(B)}, y_A) + (1 - \lambda) \cdot \ell(\mathbf{h}_A^{(B)}, y_B).$$

5. 因为神经网络的传递性，只需选取某一层（比如第一层）数据来处理，就能让效果扩散到整个模型

## 核心算法原理概述



## 示例

原图：

A



B



A的特征信息  
混杂B的矩信息



B的特征信息  
混杂A的矩信息



人像轮廓依旧是可辨识的  
→ **LABEL** 不应该改变

## 核心算法优化空间

1. 两个Hyper Parameter的调整
2. 根据正则归一方法不同, MoEx 作用于哪一层也应当调整



## 核心算法创新点及优点

1

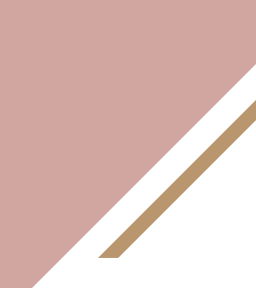

将矩（Moment）纳入模型训练

2

完全基于抽象的特征层  
具有很好的可拓展性  
(对数据类型可拓展)  
(对模型类型可拓展)

3

相比之下同类型工具  
**CutMix**  
仅仅能用于图像增扩

- 
1. MoEx比起一个具体工具更偏向于一个抽象概念，在不同的模型与不同的领域上有不同的具体实现
  2. 下文以基于PyramidNet-200的对CIFAR图像集MoEx实现为例子
- 

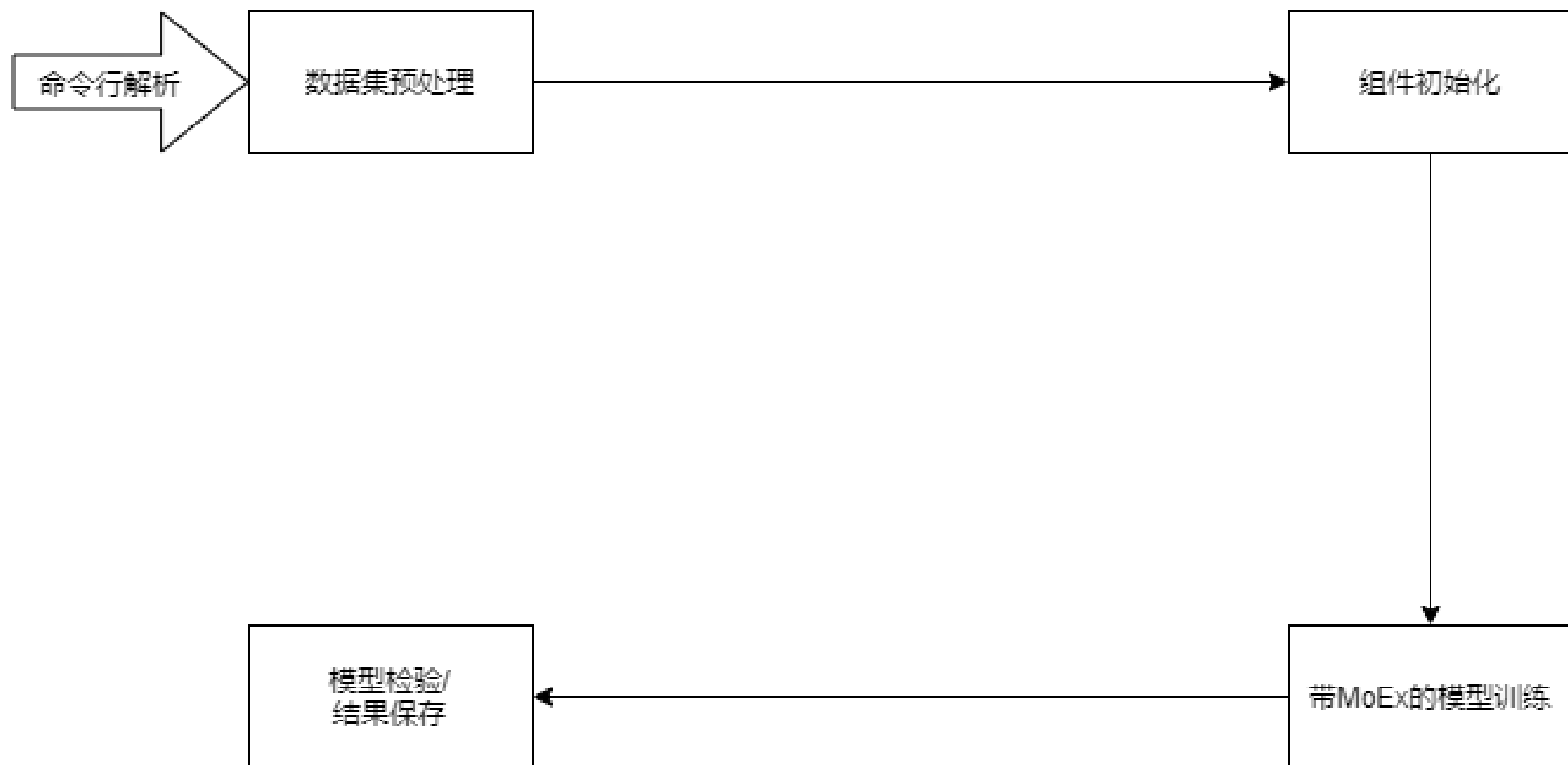


02

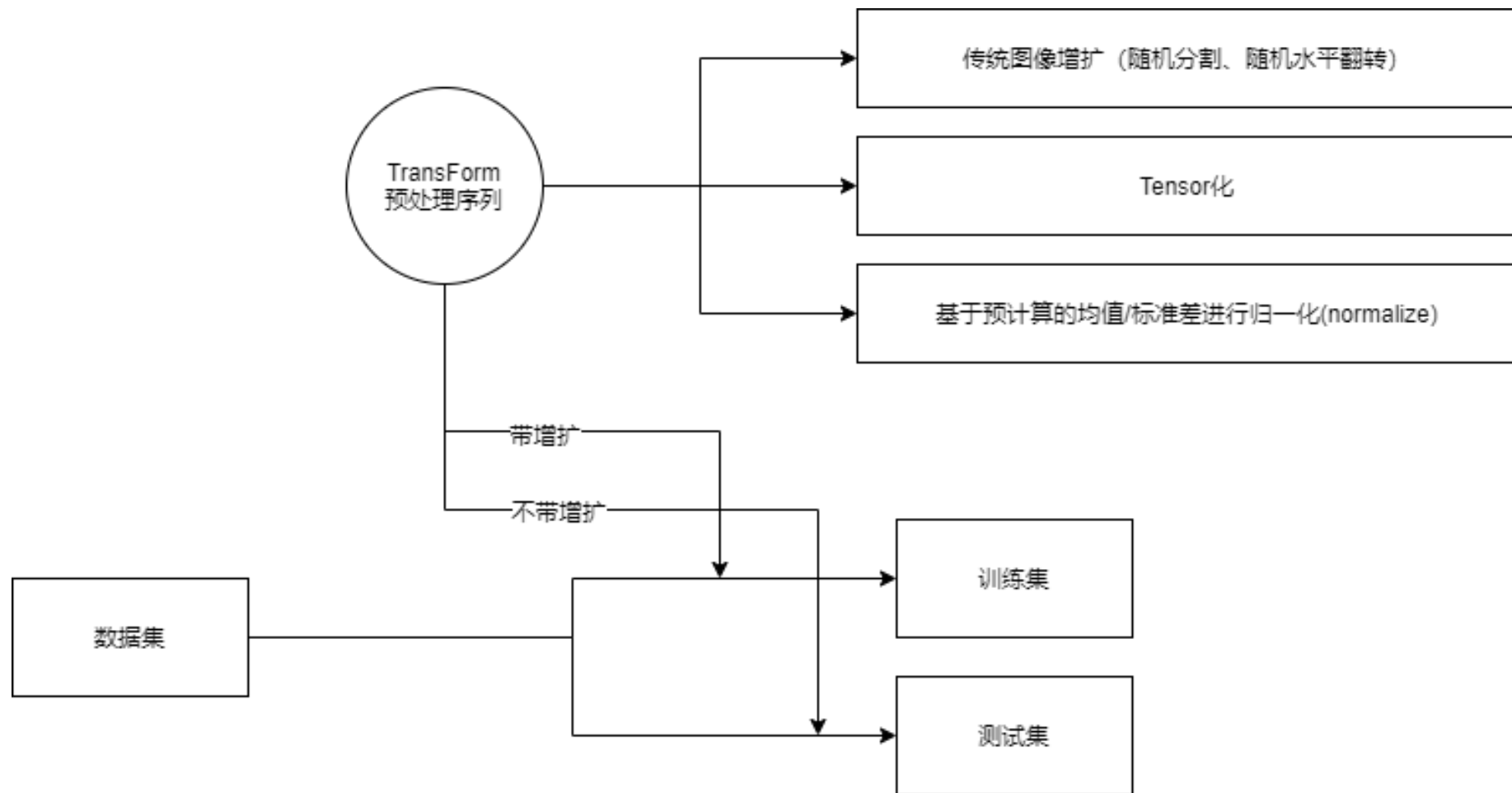
# 功能模块介绍



# 数据流概览

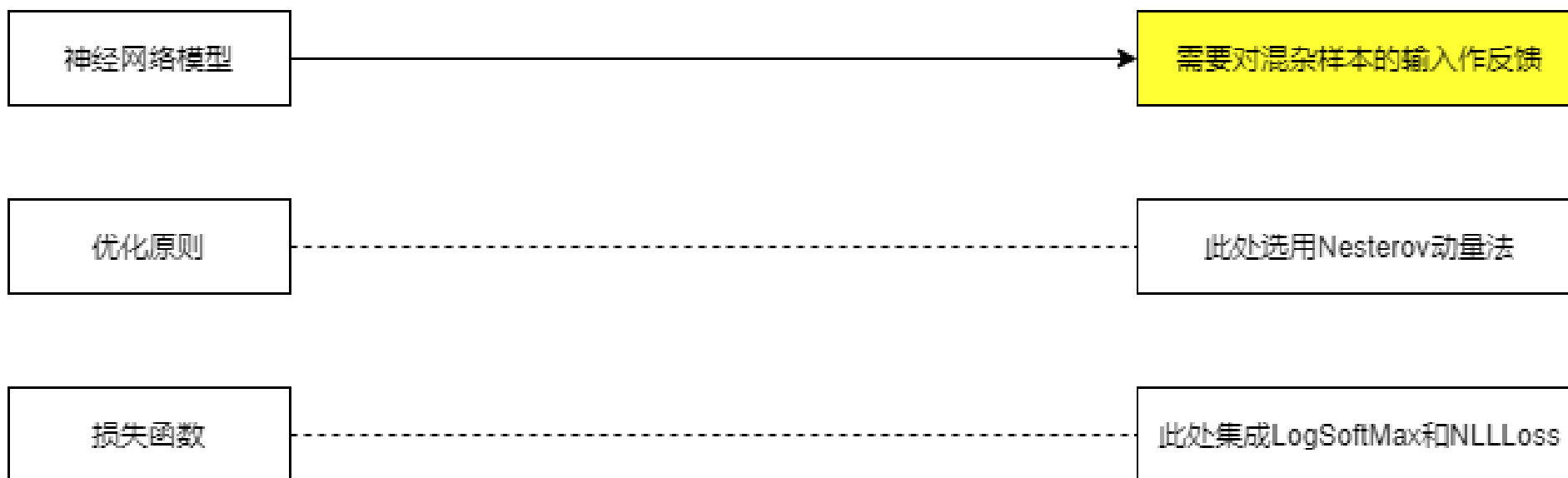


# 数据集预处理



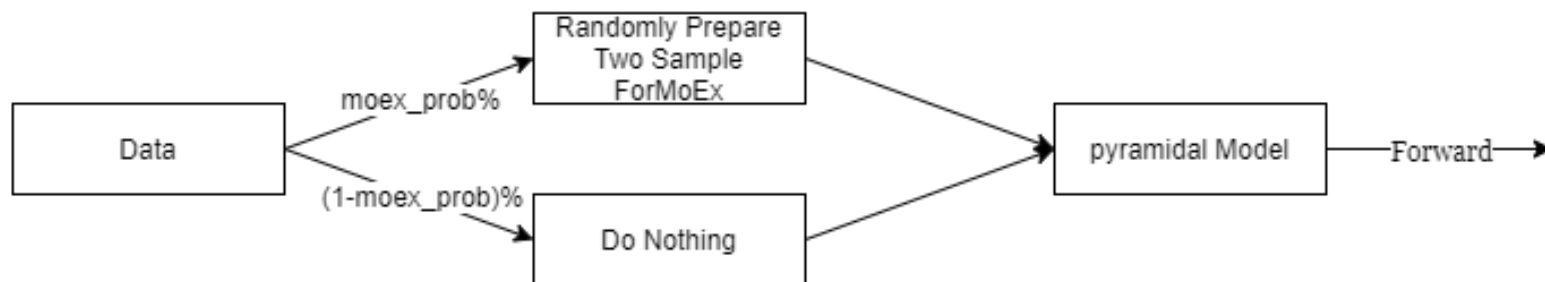
## 组件初始化

# Freely Select Your Component

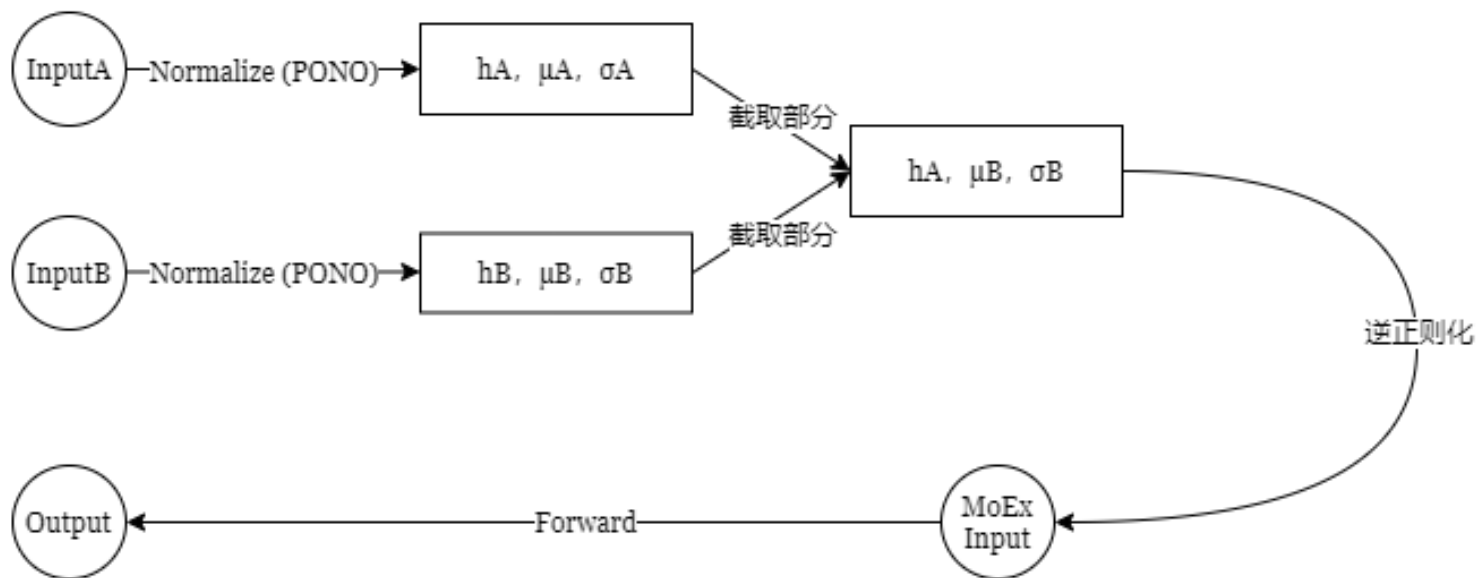


## 带MoEx模型的训练 (1)

对数据集中的每个Data



如果进行MoEx的话



## 带MoEx模型的训练 ( 2 )

$$\lambda \cdot \ell(\mathbf{h}_A^{(B)}, y_A) + (1 - \lambda) \cdot \ell(\mathbf{h}_A^{(B)}, y_B).$$

1. 以该公式为准则对损失函数进行基于MoEx发生概率的加权调整



## 模型检验/结果保存

1. 简单地使用预分割的测试集对模型正确率进行检验，并做一些结果的保存



03

# 输入输出理解



## 对于MoExDemo

输入：两张同尺寸的RGB图片，设置一张为特征图片，一张为矩信息图片

理由：尺寸相同保证混合充分，粒度适宜

输入构造：没有特别要求，如果尺寸不同可以强制缩放作为预批处理

输出：混合好的图片（详见前文示例）

## 对于基于PyramidNet-200的对CIFAR图像集MoEx实现

输入：CIFAR图集、 $\lambda$ ， $p$  两个超参数的设置，以及其他模型训练通用参数（batch\_size等）

理由：无（都是模型训练的定式）

输入构造：固定数据集输入，无需构造

输出：模型对测试集的准确率

- Top 1-err : 正确标签为匹配度最高标签的概率
- Top 5-err : 正确标签在匹配度最高五个标签中的概率
- Train-loss : 损失函数的最终值

## 对于泛化的MoEx实现

输入：

- 测试/训练数据集（数据-标签对）
- $\lambda$ ,  $p$  两个超参数的设置
- 适配好的正则化方案与学习模型
- 模型训练通用参数（batch\_size等），通用组件（损失函数、梯度优化策略）

输入构造：

- 数据一般需要根据随机抽样计算的均值和方差进行预归一化
- 正则化方案，以及对应的逆正则化实现
- 神经网络模型中在Forward抽象的对应层中增加对MoEx的适配

理由：

- 保留均值=0，标准差=1的特性有助于可拓展性提高
- MoEx抽象与泛用程度过高，比较难实现统一的实现和接口
  - 实现上只要构筑好正则化与逆正则化函数，并把它运用到模型对应的一层中即可传播影响整个模型

输出：模型对测试集的准确率

- Top 1-err : 正确标签为匹配度最高标签的概率
- Top 5-err : 正确标签在匹配度最高五个标签中的概率
- Train-loss : 损失函数的最终值



汇报结束·谢谢观赏