MathWorks®
Accelerating the pace of engineering and science

快速入门指南

使用 MATLAB 进行深度学习

Deep Learning Toolbox™ 提供用于创建、训练和验证深度神经网络的内置功能。此参考资料显示的是部分常见用例。

更多示例请访问文档：*mathworks.com/help/deeplearning/examples.html*

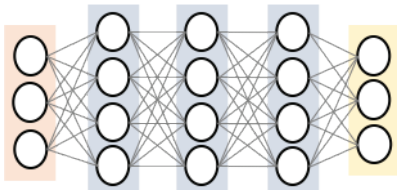
选择架构

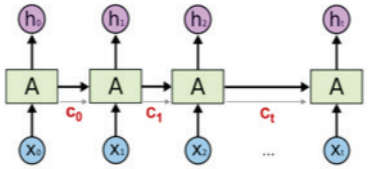
卷积神经网络 (CNN)

- 图像数据：分类，检测
- 常见层：
  - 卷积层
  - 最大池化层
  - ReLU 层
  - 批归一化层
- 从头开始训练或对预训练的模型使用迁移学习

长短期记忆 (LSTM) 网络

- 连续性数据：时序预测、信号分类、文本预测
- 常见层：
  - LSTM 层
  - BiLSTM 层
- 执行回归或分类任务





使用 **Deep Network Designer** 应用程序交互式创建和评估网络

预训练的网络

导入网络

该工具箱提供了多个用于导出模型和层的函数。更多内容请参见 GitHub 和 *FileExchange*。

导入层	`importCaffeLayers` `importKerasLayers`
导入网络	`importCaffeNetwork` `importKerasNetwork`
导出	`exportONNXNetwork`

预训练模型

从附加功能资源管理器中，使用以下其中一个命令导入网络：

`alexnet`	`vgg19`	`inceptionv3`
`googlenet`	`resnet50`	`squeezenet`
`vgg16`	`resnet101`	

训练选项

训练选项

`Execution Environment`	并行、GPU、多 GPU、自动（默认）
`MaxEpochs`	一轮训练是对整个训练数据集的一次完整遍历。
`MiniBatchSize`	训练集的子集，用于评估梯度并更新权重
`InitialLearnRate`	初始速率越高，训练速度越快，但是可能会发生偏离
`LearnRateSchedule`	随着时间按特定系数降低学习速率
`ValidationData`	在训练中验证
`ValidationPatience`	如果精确度重复几次则停止训练（节省时间）

验证

推理

`predict` 返回每个类的概率

`classify` 返回每个类的标签和概率

```
[Ypred,scores] = classify(net,X);
```

状态

网络状态可通过 `predictAndUpdateState` 和 `classifyAndUpdateState` 采集和更新

可视化

可通过 `trainingOptions` 指定多种验证和可视化形式

`Plots`	进度可视化
`Verbose`	设为 true 可显示每轮训练的进度
`VerboseFrequency`	显示频率
`OutputFcn`	自定义函数
`CheckpointPath`	每轮训练中用于保存模型的目录

改进精确度

模型精确度的改进取决于任务和数据。常见方法包括：

网络架构：

- 使用来自社区专家的预训练模型
- 更新层并调整参数

数据准备：

- 添加数据
- 训练/验证/测试拆分
- 数据归一化
- 移除异常值
- 均衡类（添加权重）

超参数调优：

- 使用贝叶斯优化进行训练参数调优
- 使用 `optimizableVariable` 设置问题
- 编写函数调用模型和选项
- 使用 `bayesopt` 执行优化

```
obj = bayesopt(ObjFcn,OptVars,…);
```

了解更多： *mathworks.com/solutions/deep-learning*

mathworks.com