**基于多深度学习网络模型融合的交通模式识别算法**

# Abstract

随着智能手机的普及和移动互联网的发展，基于移动智能设备的用户行为识别技术已经成为普适计算的热门研究领域。个人交通模式的识别作为其中一个重要的研究方向，可以为城市规划、交通管理、用户运动分析和个性化服务提供有效的数据支持。本文提出一种基于多深度学习网络模型融合的交通模式识别算法，对低功耗传感器数据进行滑动平均滤波、特征提取之后，筛选在不同交通模式下辨别度较高的特征，将特征向量分别送入DNN、CNN、LSTM中学习深度特征，并利用贝叶斯投票算法来确定最终的识别结果。实验结果表明本文所提出交通识别算法，可以针对小汽车、公交车、地铁和火车进行细粒度识别，准确率高达98%，相比传统机器学习算法有着更好的鲁棒性和泛化能力。

# Key Words

深度学习 交通模式识别 神经网络

# Introduction

近年来，随着智能手机传感器的广泛应用，基于移动传感的交通模式识别具有越来越受到研究者的关注。交通模式识别具有实时监测交通状况，提供合适的道路出行建议，解决交通拥堵，促进绿色出行，投放定制广告等重要意义。

目前有关用户交通模式识别领域的研究已涌现出许多成果。主要利用的是GPS（Global Positioning System）和加速度传感器数据，依靠SVM、HMM、AdaBoost等传统的机器学习算法对其进行分析和研究。该策略存在功耗过高、识别过程受环境影响、识别鲁棒性差等问题；同时，传统机器学习算法花费大量的时间设计手工特征，且人工设计的特征不仅需要专业领域的知识，而且往往具有一定的局限性。

深度学习作为机器学习研究中的一个新领域，是具有更多隐含层的神经网络结构，能够在数据中进行更高层次的抽象和预测**[1]**，已经在计算机视觉**[2]**、自然语言处理**[3]**、文本处理**[4]**等领域取得巨大成功。研究表明，利用神经网络从大规模的时间序列数据和图像数据中学习深层特征，表现出优异的性能**[5]**。但是在交通模式识别领域的研究成果较少，并且，现阶段基于深度学习的交通模式识别大都使用的是基础的深度神经网络（Deep Neural Network，DNN），对其他网络模型的尝试相对较少。

本文提出了一种普适的交通模式识别策略，在不受外界环境、手机放置位置的影响的情况下，细粒度识别机动交通模式。该方法基于智能手机内置的低功耗传感器，通过采集大量的用户交通数据训练神经网络模型，从而实现模式的检测。本文的主要贡献和创新之处如下：

1. 提出一种基于多源传感器数据融合的交通模式识别算法。数据源方面不仅选择了传统的加速度计，在此基础上还引入了陀螺仪、地磁、气压传感器，提高了数据层面的多样性，有助于提高检测的精度。

2. 对传感器数据进行浅层特征的提取。首先，通过减少重力对加速度的噪音作用丰富了加速度特征的提取维度；再从时域、频域、统计学等多角度提取有效特征，进一步通过绘制CDF 图，完成对多源传感器数据特征的过滤，筛选出在不同交通出行模式下表征能力较强、辨识度较高的特征。

3.设计并实现了多个轻量级神经网络模型融合的模式识别算法。本文采用了深度神经网络DNN、卷积神经网络（Convolutional Neural Network，CNN）、长短时记忆网络（Long Short-Term Memory，LSTM）3种典型的神经网络模型，从浅层特征数据中学习深层特征。其中，使用数据标准化方法加速模型收敛，采用Dropout、自适应学习率等相关算法策略对网络模型进行优化。最后根据贝叶斯投票策略进行融合，确定最终的识别结果。

# Related work

近年来，已有不少研究者针对交通模式问题提出了解决方案。

Mun等人**[6]**结合GSM基站和WIFI信号变化来区分静止、步行和驾车，平均精度达到80%-90%。GSM和WIFI仅提供对移动速度的粗略估计，适用于粗粒度的交通模式识别，对于细粒度区分机动交通工具的效果较差。

Wang 等人**[7]**对GPS数据采用稀疏自动编码（Sparse Auto-encoder SAE）、卷积神经网络和深度神经网络（Deep Neural Network，DNN）来检测用户的交通模式，利用DNN对轨迹手工特征和深度特征进行学习，获得74.1%的精度。该方案使用到了深度学习的算法和策略，但准确率较低。

Hemminki**[8]**等人利用静止分类器、步行分类器、机动车分类器组成层次分类器对加速度数据进行分析，用于区分公交车、火车、地铁、电车和汽车5种不同的机动交通模式。除方差、偏度、均方根等常见统计特征，作者还定义并提取了加速和刹车形成的峰特征以及多个峰特征组成的段特征，最终获得85%的精度。该方案使用了多种手工特征，识别精度较为良好但模型训练花费时间较长且模型鲁棒性有待提高。

Shih-Hau Fang等人利用加速度、陀螺仪、地磁传感器，通过构建DNN深度神经网络模型，针对静止、步行、骑行、跑步等用户行为以及5种机动交通模式，以离线训练，在线测试的方式进行识别，获得将近95%的准确率。并且通过与另外几种传统机器学习算法的比较，表明了该方法的有效性。

本文提出了一种基于多神经网络模型融合的交通模式识别算法，基本目标是在不依赖外部环境和设备的条件下，利用智能移动设备内置的传感器识别小汽车、公交、火车、地铁这4种最常见的机动交通模式。

本文分别使用深度学习中典型的深度神经网络、卷积神经网络、长短时记忆网络作为识别模型，为加速模型的收敛速度，对特征提取阶段所获取的特征数据进行标准化处理，输入到神经网络中进行学习。经过多次训练和迭代，调整模型获得最优的超参数，采用Dropout、梯度下降、自适应学习率等策略对模型进行优化，并对比三者的模型结构、时间复杂度、混淆矩阵和识别精确度。由于各个网络模型在不同的模式下表现不同，适用范围不一致，所以采用贝叶斯投票算法，根据各个模型的权重加权确定最终的识别结果，从而进一步提高细粒度交通模式识别的精度。

综上所述，本文的研究基于低功耗传感器和轻量级神经网络模型的交通模式识别技术，利用特征值提取来简化深度学习模型复杂度，提高模型训练效率，且提出多网络模型融合策略来提高识别的准确率。通过测试和优化，系统能够满足普适性、高精度、低功耗的要求。

# Proposed Method

本文设计的基于多深度学习网络模型融合的交通模式识别算法框架如图3-1所示：



图3-1 算法框架图

从图中可以看出，本文选择加速度、陀螺仪、陀螺仪、气压传感器作为数据源，对多源传感器的互补信息和冗余信息进行融合，经过数据预处理，提取浅层特征。由于本文研究的机动交通工具相似度较大，仅用一种神经网络进行识别可能出现误判情况，尤其是像汽车和公交车两者容易产生混淆，所以文本提出了基于贝叶斯投票算法的多网络模型融合策略，选取了DNN、CNN、LSTM 3种有代表性的神经网络分别对交通模式进行判定。最后，利用带权重的投票决定最终识别结果。本文采用离线训练、在线测试的设计思想，模型在PC端离线训练好，然后利用这3种模型的融合策略进行在线数据的测试。

## **Data Collection**

本文利用开发的Android采集器，按照既定的时间窗口、采样频率和数据格式，获取加速度、陀螺仪、地磁和气压数据。其中，本文设置采样频率为100Hz，时间窗口为2.56秒，即10ms采集一条数据，而一个样本对应256条数据。

## **Preprocessing**

一方面，本文参考了Hemminki提出的重力分量估计算法，消除重力对加速度数据的影响，进一步将加速度分解为水平分量和垂直分量。

另一方面，本文使用滑动均值滤波算法对原始传感器数据进行平滑噪声处理，滑动窗口大小为135帧。该预处理工作旨在表征用户交通行为的周期性变化和整体趋势。

## **Feature Extraction and Selection**

本文将256个样本集成为一个数据帧，使用具有50%重叠的滑动窗口生成下一帧数据。50%的样本重叠意味着使用1.28s的数据重构样本，对应产生一次预测结果，从而减少系统的响应延迟。

浅层特征的提取以数据帧为基本单位，计算出在统计、时域、频域3个不同层面上的238个特征值，进一步通过累积分布函数（Cumulative Distribution Function，CDF）来进行特征筛选。经过若干组对比实验的分析，最终确定了169个特征，作为深度学习各个网络模型的输入。本文选择的特征列表总结如表3-1所示：

表3-1 本文选择的各传感器特征值

|  |  |  |
| --- | --- | --- |
| 传感器数据 | 类别 | 内容 |
| 水平加速度 | 统计 | Mean, STD, variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS |
| 时域 | Integral, Double integral, Auto Correlation, Mean-Crossing Rate |
| 频域 | FFT DC, 1, 2, 3, 4, 5, 6 HZ, Spectral Energy, Spectral Entropy, Spectrum peak position, Wavelet Entropy |
| 垂直加速度；  原始加速度；  陀螺仪；  地磁 | 统计 | Mean, STD, variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS |
| 时域 | Integral, Double integral, Auto Correlation, Mean-Crossing Rate |
| 频域 | FFT DC, 1, 2, 3, 4, 5, 6 HZ |
| 气压 | 统计 | Mean, STD, variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS |

这些浅层特征组成的特征向量作为训练或测试的独立样本，辨识度高，降低神经网络的模型的数据处理维度。

## **Normalization**

为归纳统一样本特征数据的大小和分布，方便对数据进行综合对比评价进行特征变换，，提高梯度下降最优解的求解速度。本文选择z-score标准化（Zero-mean Normalization）对数据进行标准化处理。

根据原始数据的均值和标准差对数据进行标准化，标准化之后的数据服从均值为 0，标准差为1的标准正态分布，其转化函数为：

其中是该属性在样本数据中的均值，是标准差， 和 分别代表标准化前后的数据。

## **Deep Neural Network Architecture**

1. 网络结构

本文设计了含有3个全连接层的神经网络，其网络结构如图3-2所示。该网络的输入层有169个节点（特征过滤后得到的169个特征），输出层有4个节点（对应汽车、公交、小汽车、火车四类模式），各层之间神经元以全连接的方式相连。

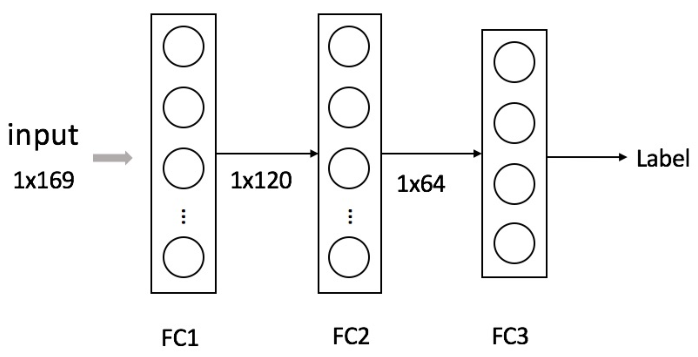


图3-2 DNN网络结构

1. 算法流程

DNN网络训练过程如算法3-1所示：

算法3-1 DNN算法流程

|  |
| --- |
| Algorithm 3-1 DNN Learning Algorithm(Shadow feature vector sf ) |
| 1： Regularization feature data, size = 169\*1 |
| 2： Repeat： |
| 3： **Forward Propagation:** |
| 4： fc1←Fully\_connected(sf, ReLu) |
| 5： droupout |
| 6： fc2←Fully\_connected(fc1, sigmoid) |
| 7： class label sigmoid(fc2) |
| 8： **Backward Propagation** |
| 9： Until loss convergences |

1. 时间复杂度

在深度神经网络中， 是第*s*层的节点数（不含偏置项），*n*是权重参数的数量，则矩阵乘法运算的总时间复杂度可以表示为：

## **Convolutional Neural Network Architecture**

1. 网络结构

图3-3是本文的卷积神经网络结构，本文采用1×169的一个样本数据作为模型的输入，为方便后续的卷积操作，将数据转换为13×13的二维数据。然后接一个二维卷积层，使用52个3×3的核去二维卷积上一层的输出，得到52个特征图，每个特征图的尺寸是11×11。在卷积层之后，使用一个最大池化层，对每个特征图使用 2×2 的采样，所以该层输出为52个5×5的特征图。接下来将数据扁平化成1×1300的一维数据，再用一个全连接层来捕获低层的特征信息，输出1×128的特征向量。最后一层通过softmax得到最终的预测输出。

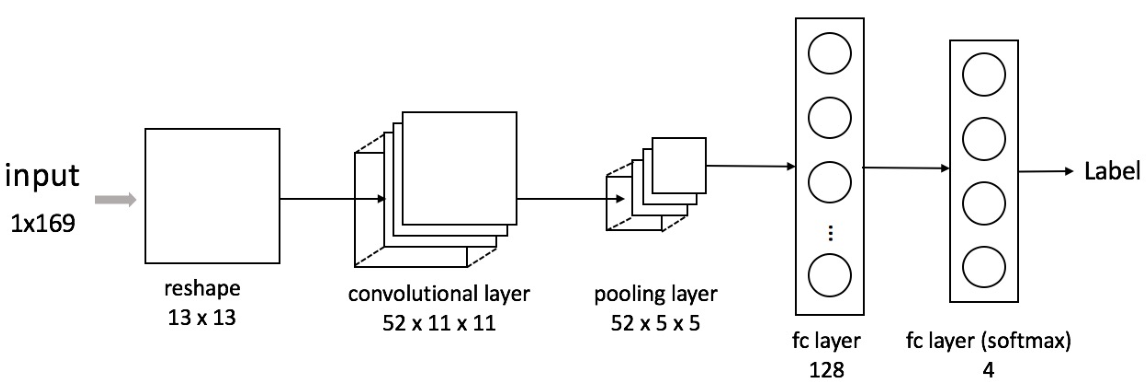


图3-3 CNN网络结构

1. 算法流程

CNN网络训练过程如算法3-2所示：

算法3-2 CNN算法流程

|  |
| --- |
| Algorithm 3-2 CNN Learning Algorithm(Shadow feature vector sf ) |
| 1： Regularization feature data, size = 13\*13 |
| 2： Repeat： |
| 3： **Forward Propagation:** |
| 4： cd ← Convoluton2D(sf, ReLu) |
| 5： mp ← Max\_Pooling(cd) |
| 6： droupout |
| 7： fc←Fully\_connected(mp, ReLu) |
| 8： class label ← softmax(fc) |
| 9： **Backward Propagation** |
| 10： Until loss convergences |

1. 时间复杂度

所有卷积层的时间复杂度为：



其中，*d*是卷积层的数量，是第*l*个卷积层的卷积核数目，同时也是第*l*层的输入通道数目，是卷积核的空间大小，是输出特征图的空间大小。

## **Long Short Term Memory Architecture**

1. 网络结构

本文设计LSTM的网络结构如图3-4所示。将标准化的169个特征作为时间序列集作为输入层，添加输出单元为50的LSTM层，再用两个全连接层捕获结果，传递给输出层。

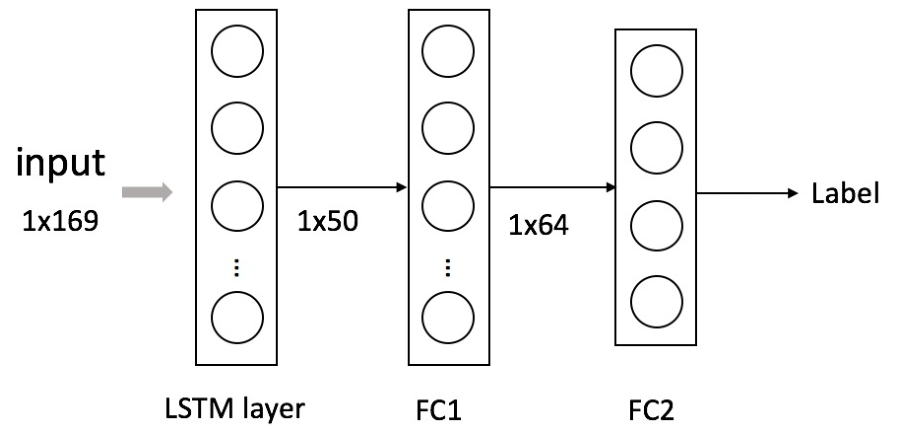


图3-4 LSTM网络结构

1. 算法流程

LSTM网络训练过程如算法3-3所示：

算法3-3 LSTM算法流程

|  |
| --- |
| Algorithm 3-3 LSTM Learning Algorithm(Shadow feature vector sf ) |
| 1： Regularization feature data, size = 13\*13 |
| 2： Repeat： |
| 3： **Forward Propagation:** |
| 4： lstm ← LSTM(sf)(dropout, recurrent\_dropout) |
| 5： fc ← Fully\_connected(lstm, sigmoid) |
| 6： class label ← softmax(fc) |
| 7： **Backward Propagation** |
| 8： Until loss convergences |

1. 时间复杂度

因为LSTM层中存在循环连接，其每个时间步内权重更新的复杂度为*O*(1)，若权重参数的数目为N，则每个时间步内训练的时间复杂度为*O(N)*。

## **Model Training and Optimization**

1. 代价函数

在模型优化中，对于误差计算，本文采用交叉熵代价函数：

其中，*y*为真实值，*a*为神经元的预测输出。参数*w*和*b*的梯度如下：

从上式可以看出，参数的更新主要受的影响，该项表示预测的输出值与真实值之间的误差。所以误差越大，梯度也就越大，参数的更新越快，训练速度也越快。

1. Adam

本文使用*Adam*优化算法进行模型优化。*Adam*算法和传统的随机梯度下降不同。随机梯度下降保持单一的学习率（）更新所有的权重，学习率在训练过程中并不会改变。而*Adam*通过计算梯度的一阶矩估计和二阶矩估计，为不同参数设计独立的自适应性学习率。

1. Dropout

图3-5是dropout的效果图，其中左图是标准的全连接神经网络，右图是使用dropout之后的网络。

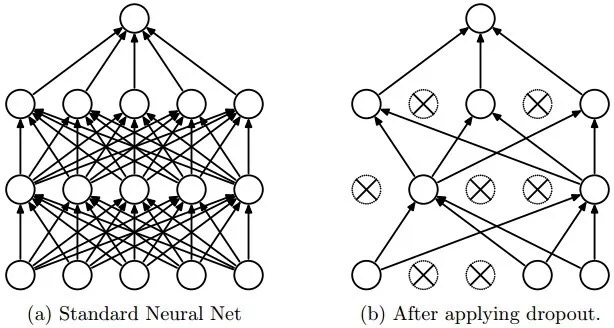


图3-5 *dropout*效果示意图

其原理为：设定一个随机概率*p，p*是超参数，表示在训练阶段随机断开神经元间的连接的比例。每次前向传播计算的时候随机让网络中该比例的隐藏层节点的权重不工作。在本次训练中不更新这部分节点的权重，只更新剩下节点的权重。而在测试阶段，使用所有节点和连接，有效防止了模型过拟合。

## **Multi Model Fusion（Voting）**

每个神经网络模型结构的不同导致了各个模型在训练过程中所关注的机动交通工具特征的角度不同，对机动交通出行模型的区分能力也不同。为进一步提高交通模式识别算法的准确率和泛化能力，对同一个测试数据，本文对各个神经网络的识别结果进行融合，确定最终的识别结果。具体的融合策略为：利用贝叶斯投票算法计算各个模型在每种机动交通模式中的加权准确率作为权重。在识别阶段，结合各个模型的识别结果和对应的权重，选择权值最大的交通模式类别作为识别结果。具体识别流程如图3-6所示。



图3-6 多模融合检测模型图

# Experimental Results and Analysis

## **Dataset and Experimental Equipment**

实验阶段中，数据采集是在设计的Android客户端采集器上进行，特征提取是在Matlab端完成，输出了特征文件。模型训练和仿真测试是在个人电脑（personal computer，PC）端基于Python工程离线完成。

数据来源为志愿者日常的采集，本次实验以100Hz的采样频率收集了50个志愿者共计400小时的交通数据。为保证数据的多样性和模型的鲁棒性，测试者的手机选用了华为、三星、小米等主流品牌不同型号的手机进行采集，实验手机均配备了加速度、陀螺仪、地磁和气压传感器。手机放置位置也可以选择为裤子口袋、上衣口袋、手中、包里等不同位置。 同时，为保证数据的多样性，增强模型的鲁棒性和泛化能力，数据采集地点包括了北京、天津、上海、杭州、深圳等城市的市区和郊区，覆盖了多种不同的道路轨迹和各种交通路况，避免了数据来源都属于同一条交通轨道所带来的模型泛化问题。

本实验所使用的特征样本数据共计230192个，数据集的具体内容如表4-1所示。

表4-1 数据集内容

|  |  |
| --- | --- |
| **Label** | **Sample Number** |
| Bus | 53778 |
| Car | 56769 |
| Metro | 61603 |
| Train | 58042 |

## **Train Process**

算法中整体模型的训练是基于深度学习框架Keras实现，其Backend采用了TensorFlow后端，实验环境Ubuntu 16.04系统，Intel i5-4440，8GB RAM，模型训练和测试使用CPU模式。

本文将表4-1划分为训练集、验证集、测试集，其比例为7：1：2。为避免实验结果的偶然性，本文随机选择训练集、验证集和测试集进行20 次实验，以多次随机实验的平均结果来评估系统的准确度。

经过实验分析，模型训练的整体批处理样本数（batch size）为128，迭代的次数（epoch）为10次，其训练效果最优。

本文分别对DNN、CNN、LSTM的实验结果进行对比和分析。

1. DNN训练过程

因为神经网络的识别能力与隐藏层的数量和每一层的神经元个数有相关关系，所以本文做出多组实验，不同DNN网络结构对应的测试交叉熵误差（Cross Entropy Error，CEE）如表4-4所示：

表4-2 不同结构的DNN训练结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入层神经元个数 | 第一个全连接层神经元个数 | 第二个全连接层神经元个数 | 输出层神经元个数 | 测试集误差  （CEE） |
| 169 | 120 | 无 | 4 | 0.0775 |
| 169 | 无 | 64 | 4 | 0.0661 |
| 169 | 64 | 64 | 4 | 0.0579 |
| 169 | 120 | 64 | 4 | 0.0327 |

本文确定第一层FC为120个神经元，第二层FC为64个神经元，两层间添加dropout，防止模型的过拟合，dropout设置为0.2，激活函数选择sigmoid函数。

图4-1分别显示了DNN在训练过程中训练和验证的精度和误差曲线，不难看出，模型在迭代10次后训练误差基本收敛，验证的精度稍有波动，整体精度在98%以上。

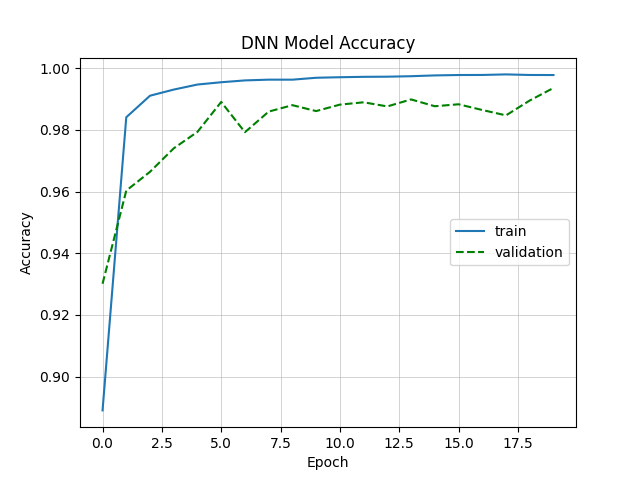
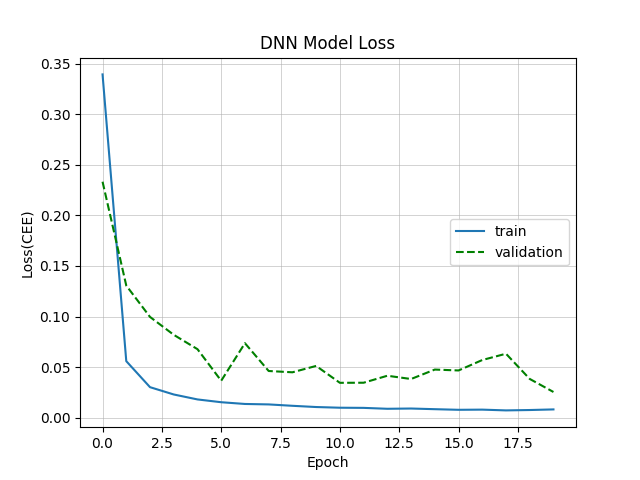
 

图4-1 DNN的训练和验证精度（左）和误差（右）

1. CNN训练过程

在卷积神经网络中为方便进行二维卷积运算，本文将输入数据转化为13\*13的类似图像的二维数据，观测不同CNN网络结构下的预测结果：

表4-3 不同结构的CNN训练结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 输入层神经元个数 | 卷积层特征图个数 | 采样层特征图个数 | 全连接层神经元个数 | 输出层神经元个数 | 测试集误差（CEE） |
| 169 | 48 | 无 | 64 | 4 | 0.0545 |
| 169 | 48 | 48 | 64 | 4 | 0.0541 |
| 169 | 52 | 无 | 64 | 4 | 0.0381 |
| 169 | 52 | 52 | 64 | 4 | 0.0379 |
| 169 | 52 | 52 | 128 | 4 | 0.0203 |

本文还分析了不同大小的卷积核对模型的影响。从图4-2中可以看出，3\*3的卷积核的大小在收敛速度和收敛效果上表现良好。

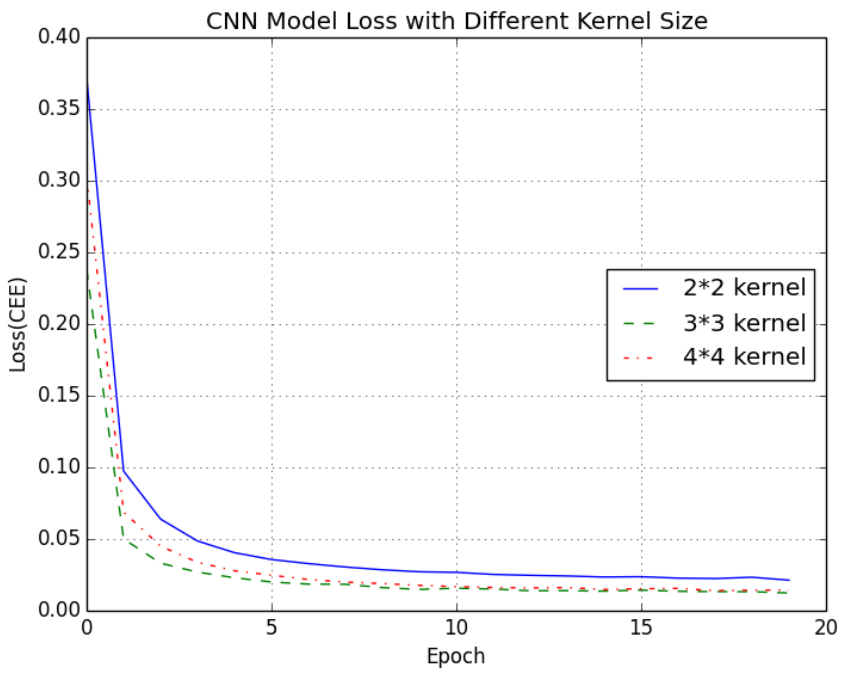


图4-2 不同大小卷积核对CNN模型误差的影响

1. LSTM训练过程

本文使用一层LSTM层构建LSTM网络，观测不同网络结构以确定最优的网络模型结构，各网络模型参数和测试误差对比如表4-4所示：

表4-4 不同结构的LSTM训练结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入层神经元个数 | LSTM层神经元个数 | 全连接层神经元个数 | 输出层神经元个数 | 测试集误差（CEE） |
| 169 | 64 | 128 | 4 | 0.0612 |
| 169 | 50 | 128 | 4 | 0.0428 |
| 169 | 50 | 64 | 4 | 0.0367 |
| 169 | 50 | 无 | 4 | 0.0595 |
| 169 | 32 | 无 | 4 | 0.0912 |
| 169 | 32 | 64 | 4 | 0.0430 |

综合表4-4的结果，本文确定输入层后的LSTM层神经元个数为50，其后接全连接网络，第一层全连接层神经元个数为64。

## **Accuracy Evaluation**

本文分别对上述提到的3种模型进行测试，得到的统计结果如图4-3所示：

图4-3 不同模型在各个交通模式下的精度

从图中可以看出，在公交车和小汽车的检测中，DNN的精确度较高，CNN对地铁的识别效果做好，而LSTM在火车的检测中精度最高。

根据多个模型的识别结果，本文采用贝叶斯投票策略进行结果的融合。最后在确定了模型在不同交通模式下的投票权重后，进行最终结果的加权投票，本文给出系统最终的检测结果，如表4-5所示：

表4-5 多模融合识别结果混淆矩阵

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实际情况  预测结果 | 公交车 | 小汽车 | 火车 | 地铁 | 精确度 |
| 公交车 | 10594 | 131 | 14 | 16 | 98.5% |
| 小汽车 | 109 | 13594 | 29 | 14 | 98.9% |
| 火车 | 64 | 53 | 18245 | 218 | 98.2% |
| 地铁 | 40 | 19 | 197 | 14763 | 98.3% |
| 平均精确度 | | | | | 98.5% |

从表中结果可以看出，对多模型识别结果进行融合之后，系统对每一类机动交通模式的识别精度都达到98%以上。其中，小汽车和公交车、火车和地铁由于机动特征相似度较高，识别容易发生混淆。

总体来看，系统平均的精度能达到98%以上。系统在最后通过绘制ROC（Receiver Operating Characteristic）图，进行AUC值（Area Under the Curve）的比较，反映出三种模型良好的分类能力。在这里给出DNN的ROC曲线图及其AUC的标注，如图4-4所示：

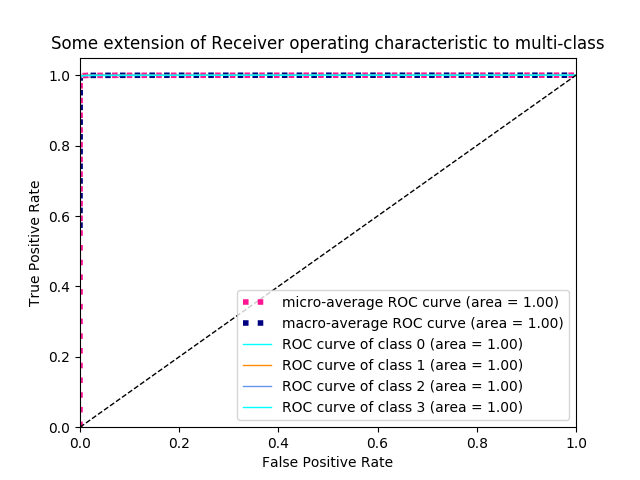


图4-4 DNN模型对应的ROC曲线和AUC值

# Conclusions

用户的交通模式识别是行为识别领域中一个重要的研究方向，在城市规划、交通管理等方面具有广阔的应用前景。本文以深度学习作为出发点，针对传统的机器学习算法区分机动交通模式存在的不足，引入神经网络模型，学习机动交通工具的深度特征，从而提高识别的精确度。

针对机动交通模式细粒度区分任务，本文分别基于深度神经网络、卷积神经网络、长短时记忆网络构建深度学习模型，并采用贝叶斯投票算法对多网络模型的识别结果进行融合，进一步提高模型的鲁棒性和泛化能力。

经过对大量数据的多次仿真测试，实验结果证明本文的算法在机动交通出行模式识别中具有 98%的精确度，能满足低功耗、高精度的需求，具有较强的普适性和稳定性。

Acknowledgment

This work was supported in part by the National Key Research and Development Program (2016YFB0502004), the National Natural Science Foundation of China (61374214), the International S&T Cooperation Program of China (2015DFG12520), and the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

Reference

[1] Lecun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521(7553):436-444.

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]. In International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[3] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6):82-97.

[4] Bo Y, Xub Z B, Lic C H. Latent semantic analysis for text categorization using neural network [J]. Knowledge-Based Systems, 2008, 21(8):900-904.

[5] Um T T, Babakeshizadeh V, Kulic D. Exercise Motion Classification from Large-Scale Wearable Sensor Data Using Convolutional Neural Networks [J]. 2016.

[6] Mun M, Estrin D, Burke J, et al. Parsimonious mobility classification using GSM and WiFi traces[C]. In Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets). 2008.

[7] Wang H, Liu G J, Duan J, et al. Detecting Transportation Modes Using Deep Neural Network[J]. Ieice Transactions on Information & Systems, 2017:1132-1135.

[8] Hemminki S, Nurmi P, Tarkoma S. Accelerometer-based transportation mode detection on smartphones[C]. In ACM Conference on Embedded Networked Sensor Systems. ACM, 2013:13.