**METIER: A Deep Multi-Task Learning**

**Based Activity and User Recognition Model Using Wearable Sensors**

1. **概述**
2. **文章的Motivation**

* **活动识别**

1. 训练好的动作识别模型通常不具有泛化能力，就是说在做实验的这批用户表现可能还可以，但是换一批用户表现就下降了
2. 所以动作识别也有两种极端，一种是**追求识别的精度**，这种采用的方法是个性化定制，为某个人或者某些人专门训练一个模型出来；第二种是**追求泛化性**，通过神经网络（比如GAN网络）提取域无关的特性，也就是提取共性，这样的模型在不同的用户群体表现都差不多，但是识别的精度就差了不少

* **用户身份识别（UR）：**

1. 身份识别也有类似的问题，在感知领域，通常进行身份识别的方法是采用步态识别，所以如果活动一变，比如现在把采集好的手势动作数据导入模型进行计算，那么准确性就大打折扣

* **Motivation：**这篇文章就想到使用多任务学习的模型，将动作识别AR和用户识别UR结合起来，在两个模型中间传递知识，相当于动作识别的时候，告诉你用户是谁，用户识别的时候，告诉你是什么动作；当然效果并没有这么明显，这样说只是为了更好的理解模型的作用

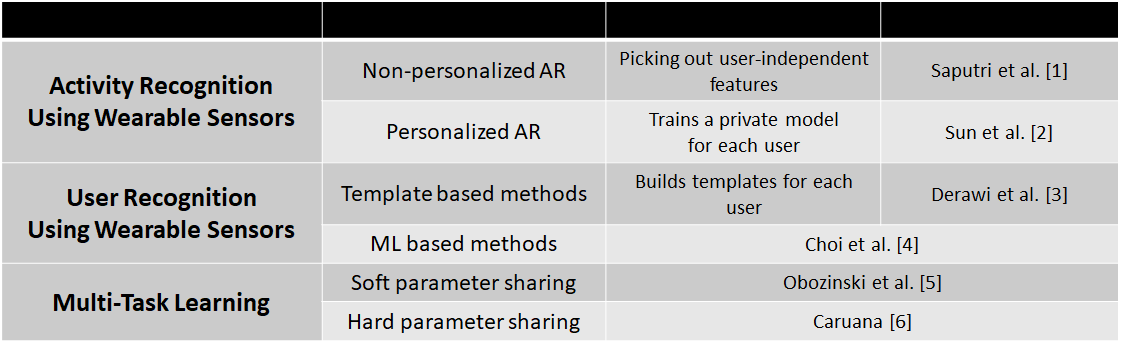
1. **核心问题**

* 从动机可以看出来这篇文章是做多任务学习的，需要融合两个模型
* 所以核心问题是**如何将两个模型融合起来**（即如何在多任务模型中间传递参数）

1. **核心方法**

* 本文**参数共享的方式为软共享**
* 其次还加上了一个**互相的注意力机制**

**二.相关工作**



**1.活动识别**

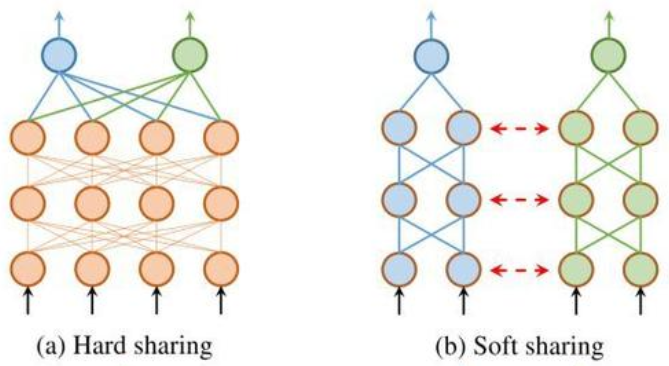
* 这里又细分为了个性化和非个性化，就是前文所讲的两种
* 而这篇文章采用的是非个性化+多任务学习

**2.用户识别**

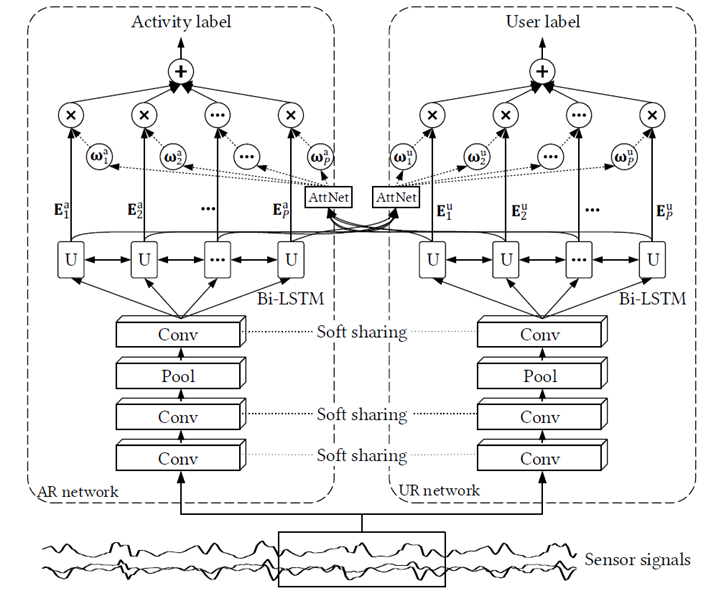
* 这里也分为两种，template based methods and ML based methods
* template based methods可以理解为采用模板的方法。是指对用户的某种信息进行提取，放到一个模板库中。以后收集到数据只需要与模板库中的模板数据进行比较，就可以知道是谁了
* 文章举了个例子：Derawi et al.[3]创建一个平均步态周期模板，之后只需要计算输入的数据与模板数据之间的DTW距离，其实即使计算相似性，距离越小相似度越高
* ML based methods最常见的就是人脸识别

**3.多任务学习**

* 这里按参数共享的方式分为软共享和硬共享，实际上参数共享的方式多种多样，这里只列举出重要的两种
* 硬共享是目前应用最为广泛的共享机制，它把多个任务的数据表示嵌入到同一个语义空间中，再为每个任务使用一任务特定层提取任务特定表示。硬共享实现起来非常简单，适合处理有较强相关性的任务，但遇到弱相关任务时常常表现很差
* 软共享为每个任务都学习一个网络，但每个任务的网络都可以访问其他任务对应网络中的信息，例如表示、梯度等。软共享机制非常灵活，不需要对任务相关性做任何假设，但是由于为每个任务分配一个网络，常常需要增加很多参数



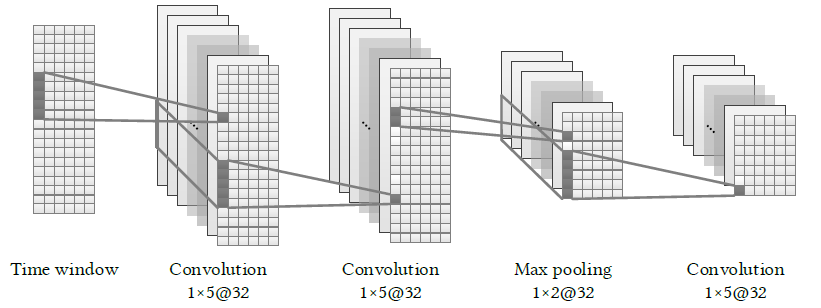
**三.模型**



**1.输入信号**

* 输入的数据是多个传感器的序列组合而成的，划分为一个个时间窗口，每个都有对应的用户标签和活动标签，也就是有两个标签，进入不同的模型与不同的标签计算损失

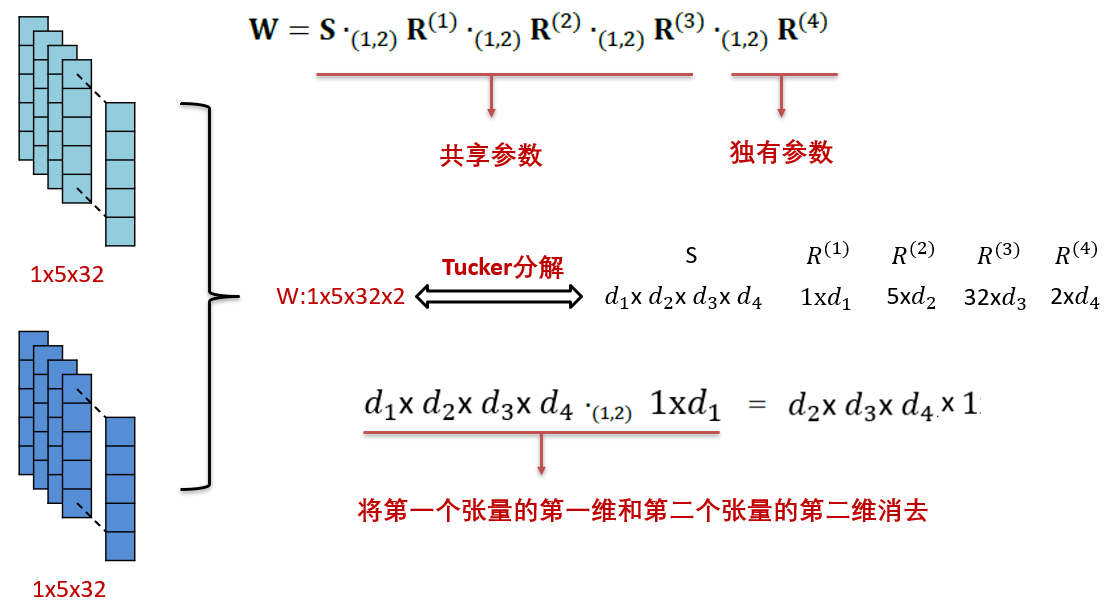
**2.CNN网络**



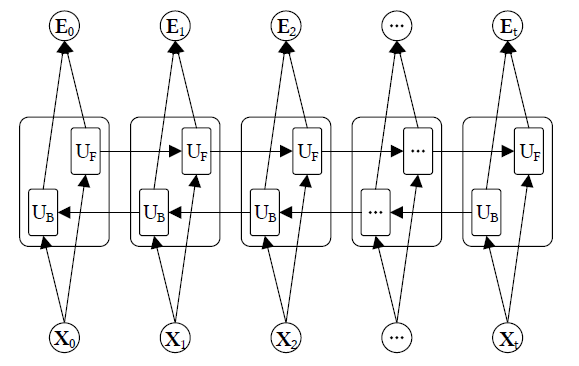
* CNN网络的作用是特征提取
* 包含卷积层和池化层：卷积层进行特征提取，池化层的作用是减少后续网络的计算量，并且防止过拟合（因为数据越详细过拟合的风险越高）
* 卷积核大小为 1x5 ，池化是1x2的范围取较大值，序列池化后缩小了一半

**3.软参数共享**

* 采用的方法是张量分解的方法
* 文章中解释到采用这个方法的原因是which can learn what and how much to share，就是说这个张量分解的方法知道分享什么参数，而且知道什么时候分享
* 张量分解的方式有多种多样，这里采用的是Tucker分解，可以类比二维矩阵可以分解为两个一维向量的乘积，高维张量就更为复杂
* 分解的对象是两个模型的卷积层合成的四维向量W，等号后面是分解的结果
* 然后在两个模型之间分享前四个张量，最后一个张量为模型独有的
* 模型的训练过程：分为两个阶段，先不进行参数分享，各训练各的模型，然后再把训练好的卷积核合成四维张量，进行张量分解，这时候模型的参数就是分解后的张量了而不是卷积核，卷积核由分解后的张量计算得到

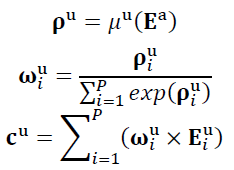


**4.Bi-LSTM**



* 因为这里的输入是一个时间序列，所以需要捕获时间上的联系，采用的是一种双向LSTM
* 需要注意的是这里CNN网络的输出需要沿着时间维度进行扁平化，使其变为一个向量

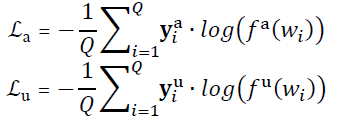
**5.Mutual Attention**



* 前面的软参数共享在不同的模型之间传递知识，这里的注意力机制可以利用一个任务的知识来突出另一个任务的重要特征
* 可以看上面的公式，就是将用户特征经过一个神经网络得到一个结果，然后标准化之后就是权重了，过程十分简单
* 值得注意的是其中的μ是一个双层的全连接层神经网络

**四.模型训练过程**

**1.单独训练时的损失函数**



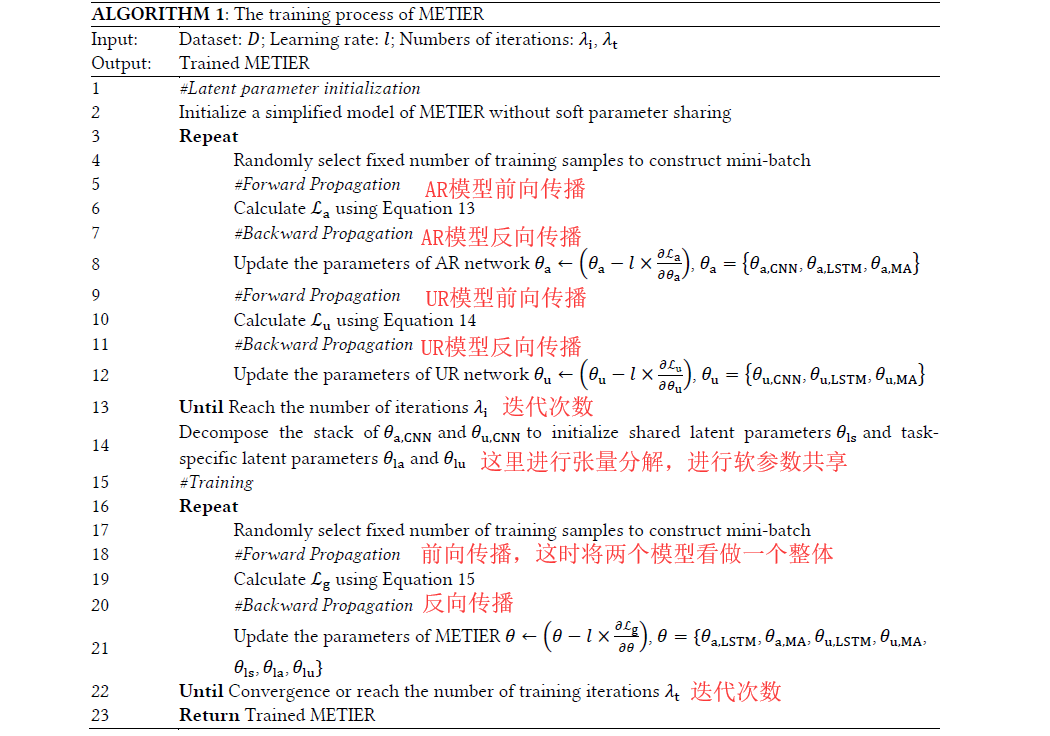
* 上面是两个模型单独运行时的损失函数
* 是输入的序列，f是模型，是实际的标签，Q是训练集中i的总数

**2.一起训练时的损失函数**



* 这时模型之间共享了参数，可以看做是一个大模型
* 是总的损失函数，​和是之前模型单独训练的损失函数，α是超参数，后面实验说是设置为1，所以可以看出两个模型的作用差不多

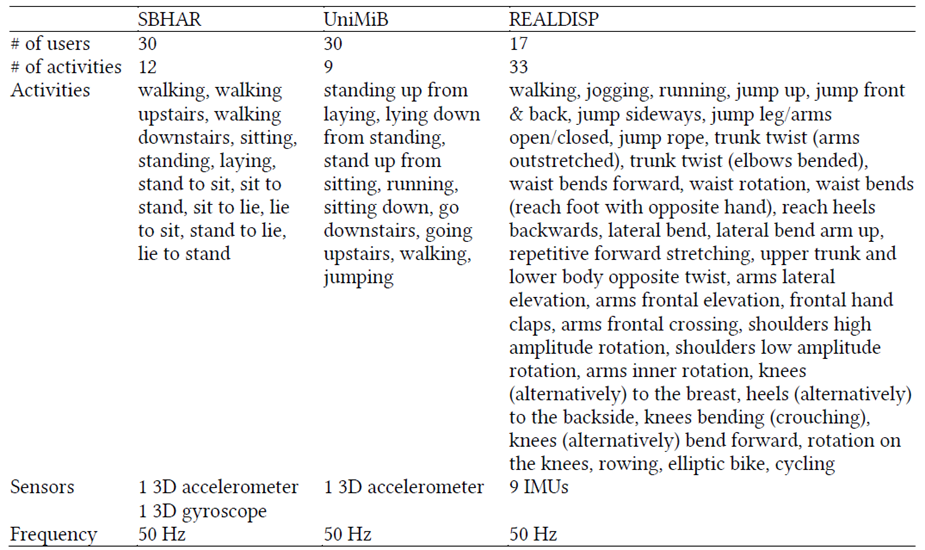
**3.整个模型训练流程**



* 可以看到也是比较简单的一个过程，先单独训练，然后共享参数，然后设置一个总的损失函数对整个模型进行训练，得到最终的训练模型

**五.实验记录**

1.首先数据集使用的是三个公开的数据集，具体细节如下表



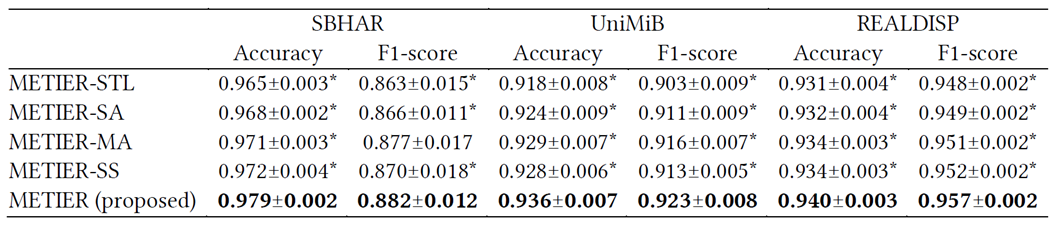
**2.学习率**

* 设置为0.0005，设置为0.003
* rs设置为2000，i是迭代次数



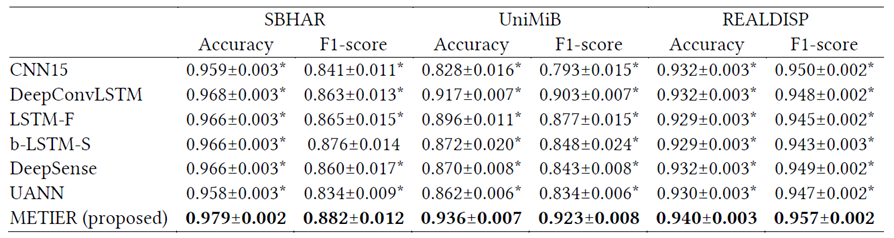
**3.与修改的模型进行对比**

* 相当于控制变量法，看每个独立的模块的实际作用如何
* METIER-STL是训练两个完全独立的模型
* METIER-SA是加上了自我注意力机制
* METIER-MA是加上相互注意力机制
* METIER-SS是加上了参数软共享



**4.与其他模型进行比较**

* 这里是与当前效果比较好的模型进行横向比较



**六.总结**

* 这篇文章还有继续探索的空间
* 参数共享模式多种多样，文中也没有给出不同参数共享模式的比较，所以后续可以尝试在这一点继续探索，可以使用最新的参数共享方式试试模型性能如何
* 另外，这里的多任务不仅仅局限于活动识别和用户识别这两类，还可以探索其他的任务之间的联系

**参考文献**

[1] Theresia Ratih Dewi Saputri, Adil Mehmood Khan, and Seok-Won Lee. 2014. User-independent activity recognition via three-stage GA-based feature selection. International Journal of Distributed Sensor Networks 10, 3 (2014), 706287.

[2] Xu Sun, Hisashi Kashima, Ryota Tomioka, Naonori Ueda, and Ping Li. 2011. A new multi-task learning method for personalized activity recognition. In Proceedings of the 11th IEEE International Conference on Data Mining. IEEE, 1218–1223.

[3] Mohammad Omar Derawi, Claudia Nickel, Patrick Bours, and Christoph Busch. 2010. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE, 306–311.

[4] Sangil Choi, Ik-Hyun Youn, Richelle LeMay, Scott Burns, and Jong-Hoon Youn. 2014. Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification. In Proceedings of the 3rd International Conference on Computing, Networking and Communications. IEEE, 1091–1095.

[5] Guillaume Obozinski, Ben Taskar, and Michael Jordan. 2006. Multi-task feature selection. Statistics Department, UC Berkeley,Technical Report 2 (2006).

[6] Rich Caruana. 1997. Multitask learning. Machine Learning 28, 1 (1997), 41–75.