厦門大學

本 科 毕 业 论 文

（主修）

基于深度学习的逆向静态动力学解算器及应用研究

Deep Inverse Kinematics Solver and Its Applications

姓　　名：李游宏

学　　号：24320152202770

学　　院：软件学院

专　　业：软件工程

年　　级：2015 级

校内指导教师：郭诗辉　副教授

二〇一九年五月六日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

另外，本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年　　月　　日

致谢

在此论文完成之际，首先感谢郭诗辉老师在立题以来一直的引导、支持、耐心、鼓励与鞭策。因为我个人的原因和学业因素的存在，论文的实验过程从 18 年初春开始，于 19 年初夏结束，前后一年有余。此期间，正如同所有的科研工作一般，困难不断。从深度学习和人体动作捕捉文件的入门问题，到输出动作序列的降噪；

他并不仅仅是我的毕业论文指导老师，更是用这个题目让我对做学术研究有了大概的了解，同时也是身体力行，让我体会到了做科研需要的严谨、勤奋、不懈。

摘要

在机器人控制和角色动画领域，逆向运动学（IK）是一个久远的问题。该问题主要的难点在于冗余性，也就是不同但可数范围内的身体形态可能都指向同一个骨骼末梢执行器的位置。从全部的代数解中选择最合适的正确解是一个悬而未决的问题。因为人们对人体运动的熟悉度以及对细微细节的高度敏感，所以对人体形态角色是否具有自然姿势的识别尤其具有挑战性。本文用迄今为止最大的人类 mocap 数据库训练的深度学习神经网络来解决逆运动学问题，定义了输入和输出的运动帧之间的临界时间相关性，并对不同参数下神经网络训练表现进行了系统的对比。鉴于逆向运动学问题中多解决方案的问题，神经网络训练后模型会选择与真实表演者的人体姿态最一致的姿势作为输出，其中的一致性通过与基准数据库进行比较来验证。根据网络实际的表现，提出去噪滤波器对预测结果进一步改善。训练后的神经网络模型可以适用于诸如篮球运球动作或者芭蕾舞的复杂的任务，以及不同几何长度的角色。本文对肢体长度不做精确的长度计算，并且不需要关节做手动的设置，这消除了个体之间的差异性并允许本文的方法直接用于姿势估计的问题。

关键词：深度学习；最大的 mocap 数据库；时间相关性；复杂任务；几何长度；姿态估计

Abstract

Inverse Kinematics (IK) is a long standing problem in the fields of robotics control and character animation. The main challenge lies in the redundancy, where an infinite number of body configurations may reach the same position of end-effector. Selecting the appropriate one from the large repertoire of candidates is an open question. It is particularly challenging to identify the natural posture for humanoid characters since we are most familiar with human motion and highly sensitive to subtle details. This paper addresses the problem of Inverse Kinematics with deep neural network, trained with so-far the largest human mocap database. We identify the critical temporal correlation between input and output motion frames and compare systematically the performance of nerual network training with different parameters. Given the challenge of multi-solution in the IK problem, the trained model selects the pose which is most consistent with the pose by the real performer. This consistency is validated by the comparison with the benchmark database. A denoising filter is proposed to further improve the prediction results based on the actual performance of the network. The trained model is adaptable to complex tasks, such as basketball dribbling and ballet dancing, and to characters of different geometrical lengths. We do not assume the knowledge of the accurate limb lengths and skip the requirement of manual setup of joint limits. This eliminates differences between individuals and allows our method to be directly used in the problems of posture estimation.

Keywords: deep learning;largest mocap dataset;temporal relation;complex tasks;geometrical lengths;posture estimation

目录

Table of Contents

第一章　绪论

1.1　引言

正向运动学（forward kinematics）指的是在给定特定角色姿势的 3D 空间中计算末端效应器位置的过程。**逆向运动学（inverse kinematics）**指的是当已知末端执行器放置在所需目标位置时，计算沿身体骨骼连接的关节方向的过程。反向运动学（IK）意在解决人形骨骼自由度难题，即已知骨骼末端的位置姿态，计算骨骼对应位置的全部关节变量。从计算机视觉和人工智能发展历史来看，在控制人形机器人和角色动画领域，逆向运动学始终是一个难以逾越的待解决问题。在机器人控制中使用该技术时最常见的场景就是机器人手臂指定末端位置抓取的待解决任务，除此之外，通过约束其其手和脚的位置，可以做到可视地展示虚拟角色以与环境交互过程。

然而，关于反向运动学的解决任务并不是一挥而就的。该问题最大的挑战在于空间冗余度[1] 问题，即在末端执行器的位置相同的情况下，可能同时存在多个关于身体姿态的解。尽管所有的解在数值上都是正确的，但某些解对应的身体姿态看起来并不自然。这一问题严重影响了图像应用（如游戏和电影）中角色动画合成的自然性。在与真实人类交互的混合环境中的社交机器人方面，该问题也是日渐显著。

我们通过使用深度学习神经网络对姿势分布建模来解决 IK 问题。神经网络的训练集包含超过 200 万个采集的真人的姿势数据和以此为依据计算得到的末端执行器位置和关节方向之间的映射。现有的数据驱动方法通过利用隐藏模型中嵌入的先验信息解决 IK 问题，包括高斯过程隐变量模型（Gaussian Process Latent Variable Model，GPLVM）[2]，主成分分析（Principal Components Analysis，PCA）[3]，多变量高斯分布模型（Multi-variate Gaussian distribution model，MGDMs）[4]。但是，这些模型都只能处理中等大小以下规模的数据[5–7]。尽管目前已经有一些使用深度神经网络进行角色控制的工作[8–10]，但我们并没有找到和我们工作一样的解决 IK 问题的应用开发。据我们所知，这项工作是首先使用相当大小的数据库训练 IK 问题解算器。更具体地说，本次毕业论文的贡献是：

- 我们优先考虑符合人体自然姿态的运动，并且更倾向于由跟踪的真人示范的自然姿态。据我们所知，本次毕业论文是最先利用这样一个大型数据库来解决 IK 问题。现有的方法深度学习技术来解决 IK 问题（特别是机器人手臂运动的 IK 问题）

- 我们定义了输入动作帧和输出动作帧之间的临界时间相关性，并将短序动作序列作为网络输入来提高预测精度。除此之外，我们将一维的均值滤波器应用在输出关节通道以达到动作稳定的目的。

- 我们使用了恒定肢体比率的假设和相对坐标。这是基于以下观察结果：股骨：胫骨的长度比和肱骨：尺骨的长度比非常相似（分别为 1.21 和 1.22），个体之间的差异很小（$\leq$ 7%）[11]。这个结果说明可以消除肢体长度的影响，因此该工作可以适用于各种身体长度。

与传统的解析方案相比较，我们的方法能够避免预测出运动奇点和不可能的姿势。因为在传统方法中，病态矩阵的反演可能导致这些情况。我们的解决方案做到了与解析方案相当的运行成本，并且比迭代求解器的表现更好。

第二章　相关工作

2.1　逆向运动学解决方案

逆向运动学问题是一个长期且重要的问题。对该问题的解决是很多技术挑战的实践基础，例如在真实环境中控制人形机器人的末端执行器[12] 和在虚拟环境中模拟动画角色动作[13]。然而，随着机器人或者虚拟角色的动作复杂性的增加，对逆向运动学的问题的解决变得越来越困难，并且计算成本也随之急剧增加。本节简要概述了三类现有对于 IK 问题的解决技术。为了得到一个对该问题更全面的了解和对一些元解决方案（Sequential Monte Carlo[14]，基于粒子的逆向运动学求解器[15]）的理解。在计算机图形学中，最新的文章[16] 对关于逆向运动学问题的解决技术可做了总结，读者可以自行参考。

解析方法　尝试着在给定链接长度、目标位置和潜在约束的情况下，通过求解三角函数来找到所有可能的解[17–18]。解析方法的求解速度很快，可以为一些并不复杂的情况提供准确的解，例如在只有有限数量自由度的 2D 平面图内。然而，当研究领域从 2D 拓展到 3D 时，解析方法就不再适用了。因为从理论上来说，在三维空间上对该问题求解，解的个数是无限的。在解决 3D 虚拟角色身体姿态的问题上，解析方法对该问题的局限性尤其明显。在机器人技术中，与求解的稳定性和准确性相比，合成运动的视觉感知的自然度应当被赋予更多的优先度，因为视觉自然度是图形应用中的关键问题，同时也是一项复杂艰巨的待解决任务。

数值方法　通过迭代计算损失函数使之最小化来求解该问题。其中，损失函数通常表示为当前姿势和目标姿势之间的偏差。典型的技术包括 Jacobian, Newton，启发式方法等方法。雅可比方法的基本形式是计算关节的全局位置与角度参数之间的偏导数[19]。雅可比方法可以进一步细分。具体取决于雅可比行列式的形式，包括转置雅可比行列式[20]，伪逆雅可比行列式[21]，最小二乘阻尼[22]，奇异值分解[23] 等方法。基于牛顿法的方法通过使用拟合牛顿方法[24–25] 的函数的二阶近似来找到解。启发式方法包括循环坐标下降

法（CCD）[26] 和 Forward and Backward Reaching IK（FABRIK）[27]。迭代并收敛到最优解的这一过程非常耗时，所以数值方法可能不适合需要实时响应的应用程序。虽然得到的最优解在数值上是正确的，但无法保证呈现出的姿势在视觉上是自然流畅的的，或者说与真人表演者保持一致。

数据驱动方法 利用大型数据库并使用预先训练的模型来推断在已给定末端执行器位置情况下人体最有可能的姿势。数据库既可以从真实表演者[4] 捕获，也可以通过模拟生成（只适用于操作机器人）[28]。研究人员使用径向基函数来修改原有样本中的运动和位置，并向 IK 控制控制器提供情感，困难等级，健康等描述性特征[29]。多变量高斯分布模型（MGDM）被提出并用来精确地指定运动骨架的软关节约束和产生更高的精度和的稳定性[4]。在一项工作中，IK 问题被公式化为约束优化问题，并被用来解决使用主成分分析（PCA）或概率 PCA（PPCA）[3,30–32] 技术构建的潜在空间。针对不同类型的 IK 问题使用基于人体姿势的实时逆运动学系统的学习模型[2] 也令人印象深刻。该系统能够产生任何姿势，但更倾向于产生与训练数据中的姿势最相似的姿势。它可以应用于交互式角色构成，轨迹关键帧，具有缺失标记的实时动作捕捉，基于 2D 图像的姿态估计等。

深度神经网络 的出现为 IK 问题的解决带来了一定的希望。神经网络可以用于构建从全局坐标到局部联合自由度的底层映射。例如，研究人员用前馈神经网络的应用来解决威勒平面机械手中蕴含的 IK 问题[33]。除此之外，另一项研究工作利用多层感知器（MLP）和反向传播训练算法的方法，证明了在机器人控制中使用逆几何模型（IGM）时能有效降低了计算复杂度[6]。然而，在先前的研究中，特别是在处理复杂的结构或大量的训练数据的时候，研究人员使用的基于梯度的学习算法可能导致非常缓慢的训练过程。为了解决这个问题，研究人员提出了一种称为极限学习机（ELM）的学习算法[7]。该算法随机选择输入权重，再分析和确定单个隐藏层前馈神经网络的输出权重。在这一成果的基础上，研究人员提出了一种新型的递归神经网络控制器，用于对 IK 问题的神经网络学习过程进行控制和维护[34]。他们的工作采用了 Reservoir Computing[35] 和 Extreme Learning Machines（ELMs）[36] 的思想，在反向传播的过程中做了一些处理，进而降低了误差。另一项工作在深度神经网络的基础上，通过近似目标轨迹得到的实际例子来产生

新的运动序列[37]。然而，这些方法都基于一定的高级约束，因此它们不能够产生新的姿势或满足新的约束。上述研究都建立在虚拟环境中机器人和真实环境中机器人的能够准确匹配的理想化假设下。最近提出了一种基于监督学习的方法来解决机器人制造和装配过程中错误[38]。具体方法是通过比较具有和不具有未对准关节的逆向运动学函数的误差，进而可以观察到对于神经网络，未对准的情况下不会对结果产生偏差。

2.2　人工神经网络与深度学习

人工神经网络 [39] 神经网络本身不是一种用于解决特定问题的算法，它是许多不同机器学习算法的框架，负责协同工作并处理复杂的数据输入等任务。网络通过分析输入的样例来“学习”需要执行任务，通常编程过程中不用任何特定任务规则。例如，在图像识别中，网络可以通过分析已经手动标记为“猫”或“没有猫”的示例图像来学习识别包含猫的图像，并使用结果来识别其他图像中是否包含猫。网络是在没有任何关于猫的先验知识（例如，猫游毛皮，尾巴，胡须和特有的面孔）的情况下进行训练的。相反，它们会自动从它们处理的学习资料中生成待识别特征。

人工神经网络通过神经元之间的互相连接，类比于生物大脑中的突触。其可以将信号从一个神经元传递到另一个神经元。神经元接到上一个神经元的信号后神经元可以处理它，然后发信号通知与之相连的全部神经元进行信号处理。

神经元与神经元可以传输信号。该信号大多为实数。人工神经元可以具有阈值，只有收敛函数的值超过时才将信号传递给下个神经元。每个人工神经元的输出是通过一些非线性函数计算其输入得到的。各个神经元之间的关联称为“edge”。人工神经元和边通常随着训练过程调整权重，进而通过权重的增加或减少来调整连接处信号的强度。通常，不同的层可以对其输入执行不同类型的转换。层间的信号可能在多次遍历各层之后才能从第一层（输入层）传播到最后一层（输出层），即在层间进行了多次迭代。

人工神经网络方法的设计模式是以类人脑的方式思考问题、解决问题。然而，随着计算机技术和机器学习技术的发展，神经网络的注意力转移到具体的应用解决上。人工神经网络已经用于各种任务，包括计算机视觉，语音识别，机器翻译以及医学诊断等方面。

人工神经网络组成 [39]

- **神经元**带有标签 j，接收到从前任神经元传过来的输入 $p_j(t)$ 的神经元由以下部分组成：
  - 激活 $a_j(t)$, 一个神经元是否被激活取决于离散时间参数
  - 可能是一个阈值 j，除非学习函数发生改变，不然其值保持不变
  - 激活函数 f 计算从 $a_j(t)$ 开始的 $t+1$ 给定时间内的新激活、θ_j 和一个新的输入 $p_j(t)$ 用来对联系
    $$a_j(t+1) = f\left(a_j(t), p_j(t), \theta_j\right) \tag{2.1}$$
    产生一个新的增加
  - 输出函数 f_{out} 用来对激活 $o_j(t) = f_{out}\left(a_j(t)\right)$ 产生一个新的输出
- **连接、权重和偏移量**神经网络由连接组成，每个连接将神经元 i 的输出传递给神经元 j 的输入。在某种意义上，i 是 j 的前任神经元，j 是 i 的继承。每个连接都分配一个权重 w_{ij}。有时将偏移项加到输入的总权重中，作为转移激活函数的阈值。
- **传播函数**计算由前任神经元 $o_i(t)$ 的输出得到的神经元 j 的输出 $p_j(t)$，一般有如下形式：
  $$p_j(t) = \sum_i o_i(t) w_{ij} \tag{2.2}$$
  。当该函数添加偏移量时，上述形式变为以下形式：
  $$p_j(t) = \sum_i o_i(t) w_{ij} + w_{0j} \tag{2.3}$$
  其中 w_{0j} 是偏移量。
- **学习规则**是修改神经网络的参数的规则或算法，以便在网络给定输入后以产生准确的输出。通常情况下，这种学习过程相当于修改神经网络内变量的权重和阈值。

神经网络学习 [39] 在解决一个具体任务或者一类功能 F 的情况下，神经网络的学习过程指的使用一组观察值来寻找用于解决特定任务的最佳方法 $f^* \in F$。

这个过程需要定义一个损失函数：$C : F \to \mathbb{R}$。在此基础上可以寻找最佳解决方案 f^*, 即 $C(f^*) \leq C(f) \forall f \in F$。需要指出的是，“最佳”的意思是没有任何一个解决方案的损失低于最佳解决方案的损失。损失函数 C 是学习过程中的一个重要概念，因为它衡量特定解决方案距离待解决问题的最优解决方案的距离。学习算法通过搜索解决方案空间以找到具有最小损失的函数。

对于依赖于数据的神经网络训练过程，损失函数必须是观测值的函数，否则训练出的模型将与数据无关，它通常被定义为只能进行近似的统计量。举个简单的例子，例如寻找最小化 $C = E[(f(x) - y)^2]$ 的模型 f 来从某种分布 D 中提取数据对 (x, y)。在实际情况中，我们只有来自分布 D 的 N 个样本，因此，对于上面的例子，通常做法只会最小化式子

$$\hat{C} = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 \tag{2.4}$$

通过这样的方式做到在数据样本最小化损失函数而不是在整个分布上。

反向传播 [39] 反向传播算法是神经网络训练中的一种算法。神经网络是基于感知机的扩展。反向传播更新权重可以通过随机梯度下降方法来实现，如式2.5所示。

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial C}{\partial w_{ij}} + \xi(t) \tag{2.5}$$

其中，η 是学习率，C 是损失函数，$\xi(t)$ 是一个随机变量。损失函数的选择取决于学习类型（监督，无监督，强化等）和激活函数等因素。常见的激活函数和损失有 **softmax** 函数和交叉熵函数。**softmax** 函数的定义如下：

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)} \tag{2.6}$$

其中，p_j 表示该类的可能性（神经元 j 的输出），x_i 和 x_k 分别表示来自同一层的神经元的总输入。交叉熵的定义如下：

$$C = -\sum d_j \log(p_j) \tag{2.7}$$

其中 d_j 表示对于输出单位 j 和 p_j 的目标概率，p_j 是 j 的应用激活函数之后的概率输出。

学习模式的分类 [39] 目前有三个主要的学习模式——监督学习、非监督学习和强化学习。本文研究的问题属于监督学习。

- **监督学习**使用一组数据对 $(x, y), x \in X, y \in Y$，目的是在与目标函数匹配的函数类中找到一个函数 $f : X \to Y$。损失函数与映射和数据之间的不匹配有关，它隐含地包含有关问题域的先验知识。常用的损失函数是在所有的数据对上最小化网络输出 $f(x)$ 和目标值 y 之间的平均误差。使用称为多层感知机的神经网络的使用梯度下降来最小化该损失函数，进而进行反向传播训练神经网络。属于监督学习范围的任务有模式识别问题（也称为分类问题）和回归问题（也称为函数逼近问题）。监督学习也适用于序列类的数据（例如，手写任务，语音任务和手势识别任务）。

- **非监督学习**中，给定一些数据 x 并且使损失函数最小化，其可以是关于数据 x 和网络输出 f 的任何函数。损失函数取决于特定的任务（模型域）和先验假设（模型的隐含属性，其参数和可观察到的变量）。假设一个简单的模型 $f(x) = a$, 其中 a 是常数，损失函数 $C = E\left[(x - f(x))^2\right]$。最小化损失产生的 a 等同于数据的平均值。在压缩任务中损失函数可能与 x 和 $f(x)$ 之间的信息传递有关。在统计中，损失函数可能与给定数据的模型的后验概率有关。非监督学习应用的任务范围一般是估计问题; 具体应用的问题包括聚类问题，分布估计问题，压缩问题和滤波器问题。

- **强化学习**中，通常给定输入数据 x，而是由智能体与环境的交互生成数据。在每个时间点 t，智能体执行动作 y_t，并且环境根据一些（通常未知的）动态生成状态 x_t 和瞬时损失 c_t。学习目的是发现用于选择最小化某些长期损失度量的动作的策略，例如预期的累积损失。环境的动态和每项决策的长期成本通常是未知的，但可以估算得到。细化计算流程，环境通常被建模为马尔可夫决策过程（MDP）：定义状态为 $s_1, \ldots, s_n \in S$，动作为 $a_1, \ldots, a_m \in A$。同时定义概率分布如下：瞬时损失分布 $P\left(c_t|s_t\right)$，状态观测分布 $P\left(x_t|s_t\right)$ 和转换因子 $P\left(s_{t+1}|s_t, a_t\right)$。决策被定义为在给定观察的动作下的条件分布。

深度学习 [40]（也称为深层结构学习或分层学习）是机器学习的一个分支，是一种以人工神经网络为架构，对数据进行学习的算法。观测值可以运用多种方式来表示。而使用

这些更容易从实例中学习任务。

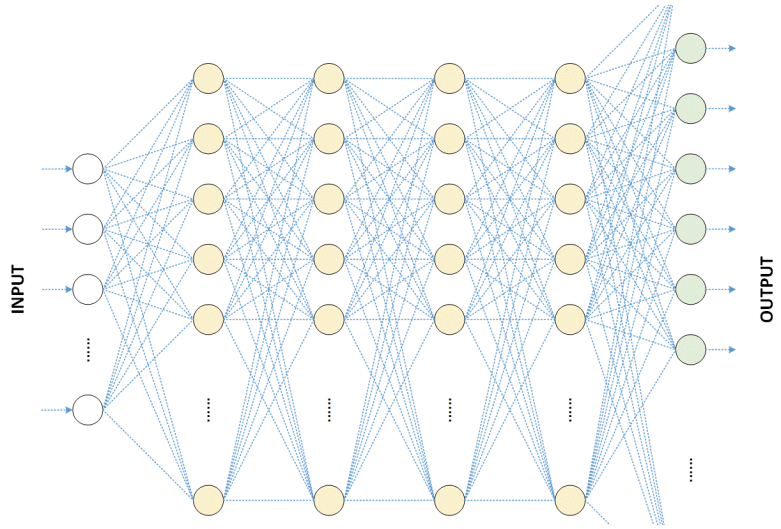


图 2.1：DNN 网络模型

深度学习框架 [40] 至今已有数种深度学习框架。包括深度神经网络、卷积神经网络和深度置信网络和递归神经网络。深度学习在计算机视觉、自然语言处理等方面已经取得了极好的效果。本文涉及的问题属于计算机视觉领域，深度学习的优越性正是选择的初衷。

深度神经网络是一种具备至少一个隐层的神经网络。深度神经网络具有更多网络层, 如图 2.1示例，一般来说第一层是输入层，中间的层数为隐藏层，最后一层是输出层。

递归神经网络 [41]（Recurrent Neural Network,RNN）是一种节点定向连接成环的人工神经网络，属于深度学习网络的一种。

- **编码器**递归神经网络将输入序列 $\vec{x}$ 编码为一个固定长度的隐藏状态 $\vec{h}$(以自然语言处理为例)
  - $\vec{x} = (x_t, \ldots, x_1)$ 是输入序列。
  - $\overrightarrow{h_t} = f\left(x_t, \overrightarrow{h_{t-1}}\right)$ 是随时间更新的隐藏状态。之前的状态 $\overrightarrow{h_{t-1}}$ 转换为和当前输入 x_t 相关的 $\overrightarrow{h_t}$。

  其中，计算隐藏状态的方程 $f(x, h)$ 是一个非线性方程，可以是复杂的 **LSTM** 单元（Long-ShortTerm Memory）。获得隐藏状态序列后，做出下一步预测：

- $p(y_t) = p(y_t|y_{t-1}, \ldots, y_1)$, 其中 y_t 是第 t 个位置上的输出。
- 以上概率可以通过隐藏状态来计算：$p(y_t) = g\left(y_{t-1}, \overrightarrow{h_t}, \vec{c}\right)$, $\vec{c}$ 是所有隐藏状态的编码，总含了所有隐藏状态。因为隐藏状态 t 就编码了第 t 个输入前全部的输入信息，y_t 也迭代式地隐含了之前的全部输出信息。

- **解码器**神经网络可以添加编码器作为解码器（$Decoder$）。编码后（$Encoded$）的信息通过解码器翻译为人类熟悉的信息。也就是上述例子中的 $y_t = f(y_{t-1}, h_t, c)$ 过程，当中非线性模型 f 就是作为输出的复发神经网络。同时需要对 h'_t 继续进行迭代：
  - $h'_t = g(h_{t-1}, y_{t-1}, c)$, $\vec{c}$ 是解码器传递给编码器的参数，是解码器中状态的 summary。h'_t 是解码器的隐藏状态。y_t 是第 t 个输出。
  - 当输入仍为 $\vec{x} = (x_t, \ldots, x_1)$，输出是 $\vec{y} = (y_t, \ldots, y_1)$，最大化条件概率 $P(\vec{y}|\vec{x})$ 就是最好的结果。

长短期记忆 [42]（$Longshort - termmemory$, LSTM）是人工递归神经网络（RNN）的一种模式，广泛地运用于深度学习领域。与标准的前馈神经网络不同，LSTM 具有反馈机制。LSTM 不仅可以处理单个数据点，也可以处理整个数据序列。

一个通用的 LSTM 单元由单元，输入门，输出门和遗忘门组成。单元负责记住任意时间间隔内的值。三个门控制进出单元的信息流。

LSTM 非常适合于对时间序列数据的分类任务、处理任务等。因为在时间序列中，重要事件之间可能存在未知的滞后和联系。LSTM 提出的初衷就是为了处理在训练传统 RNN 时可能遇到的梯度爆炸和梯度消失问题。LSTM 的优势还在于该网络对于间隙长度的不敏感性。

- LSTM 架构有很多种。常见的 LSTM 架构由单元（LSTM 单元的存储器部分）和 LSTM 单元内部信息流的三个“调节器”组成：输入门，输出门和遗忘门。一些变形的 LSTM 单元可能不具有一个或多个门，或者可能具有其他类型的门。单元负责跟踪输入序列中的元素之间的依赖性。输入门控制新值流入单元的程度，忘记门控制值在单元中保留的程度，输出门中的值用于计算输出的程度从而激活 LSTM

单元。LSTM 门的激活功能通常是逻辑功能。

- 具有遗忘门的 LSTM 单元的正向通过的形式如下：

$$
\begin{aligned}
f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\
i_t &= \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \\
o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \\
h_t &= o_t \circ \sigma_h\left(c_t\right)
\end{aligned}
\tag{2.8}
$$

其中初始值为 $c_0 = 0$ 且 $h_0 = 0$ 和运算符 $\circ$ 表示 $Hadamard$。下标 t 为时间步长。式子中变量定义如下：

  - $x_t \in \mathbb{R}^d$：LSTM 单元的输入向量。
  - $f_t \in \mathbb{R}^h$：遗忘门的激活向量。
  - $i_t \in \mathbb{R}^h$：输入门的激活向量。
  - $o_t \in \mathbb{R}^h$：输出门的激活向量。
  - $h_t \in \mathbb{R}^h$：隐藏状态向量，也称为 LSTM 单元的输出向量。
  - $c_t \in \mathbb{R}^h$：单元状态向量。
  - $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}$ 和 $b \in \mathbb{R}^h$：权重矩阵和偏差参数。

式子中的激活函数如下：

  - σ_g：$sigmoid$ 函数。
  - σ_c：双曲正切函数。
  - σ_h：双曲正切函数。

第三章　方法

在本节中，将会完整阐述本文提出的基于监督学习方法的 IK 解算器。

图 3.1说明了我们方法的整体流程。CMU 的动作捕捉库中的 BVH 文件是训练集和测试集的来源。在 MATLAB 中，对 BVH 文件进行解析和整合后，提取出神经网络的训练集和测试集。训练集和测试集一共三组，分别对应 **MP** 和 **PE** 和去噪滤波器（具体定义见3.1.2），六个文件，总大小超过 10GB。**MP** 和 **PE** 训练后，通过预先训练好的去噪滤波器，进一步对预测的姿态进行修正，得到最终结果。具体应用 **IK** 解算器时，输入的轨迹（向量序列）依次经过 **MP** 和 **PE** 和去噪滤波器，得到最终的预测姿态。

本文方法的关键组件是 Motion Predictor（MP）和 Posture Estimator（PE）。前者根据当前运动轨迹来预测末端效应器的未来位置，而后者基于一系列末端效应器位置估计身体姿势。同时加入了去噪滤波器对网络的原始输出姿态进行修正，在很大程度上使预测的姿态更自然、更流畅。

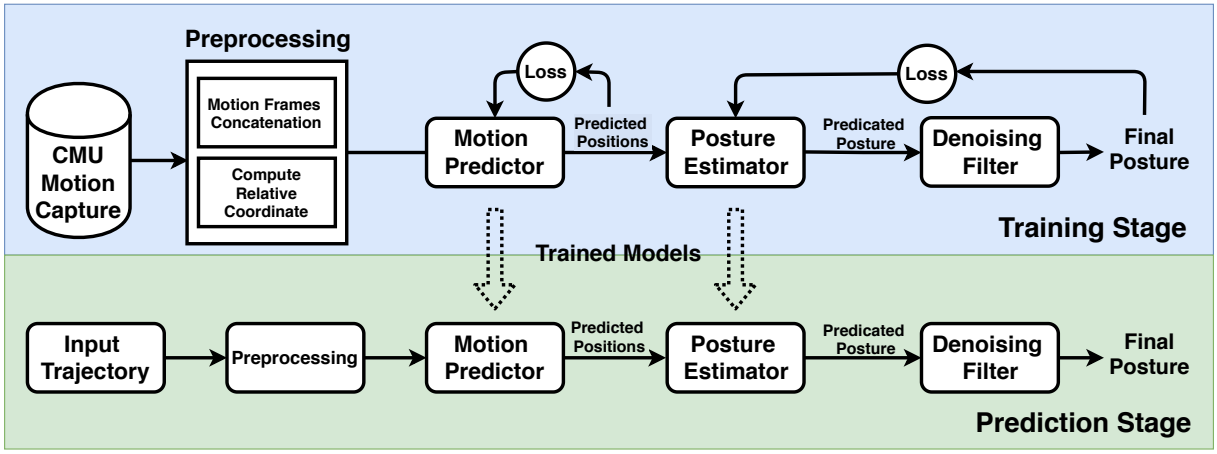


图 3.1：本方法的流程图

首先对本文中使用的一些术语进行定义：

- **骨骼（skelton）**骨骼 **S** 定义具有根关节及其相互连接的关节角色的层次结构，而关节距离适用于构造具有几何变量的角色。

- **末端执行器（end-effector）**末端执行器 **EE** 通常指与环境进行交互的角色的脚和

手。

- **姿态（posture）**姿态 **P** 是描述角色身体结构的向量，由根关节的全局变换和关节方向的角度值组成。
- **动作（motion）**动作 **M** 是一个矩阵，用于描述时域中的一系列角色姿势。矩阵行数对应帧数，列数与关节通道数成比例。
- **位置（position）**位置 **PE** 描述 **EE** 的全局位置。

3.1　离线网络学习

3.1.1　数据集和预处理

本次毕业论文使用 Carnegie Mellon University（CMU）的 Mocap 数据库来构建此问题中的神经网络的训练集和测试集。训练数据集包含约 2000 个动作序列（约 200 万个骨骼姿势），而测试数据集包含约 200 个动作序列（约 20 万个骨骼姿势）。运用 MATLAB 构建训练集和测试集时，使用正向运动学技术来计算骨骼关节的世界坐标位置。我们提出了一种分层结构，将人体的四肢假设为互相独立的。所以整个人体姿态预测可以由四个独立的 IK 解算器组成，分别用于计算左/右臂和左/右腿。

我们定义了两种有效的技巧用来提高神经网络的学习性能：

建立连续帧之间的时间关联性过往的实验中，在解决 **IK** 问题时，往往只是将其归类为一个复杂的数学问题，忽视了它也是一个运动学问题，具备一定的时间关联性。实验证明，将连续的多帧的运动序列 $[\cdots, X_{t-2*k}, X_{t-k}, X_t, X_{t+k}, X_{t+2*k}, \cdots]$ 替代传统的以末端执行器的当前位置作为网络输入在提高预测精度方面是有效的。这在我们的实验中得到了证实（见图 4.7和章节4.3.2）。

运用肩关节-腕关节/髋关节-踝节的相对坐标在本次毕业论文中，通过计算肩关节-腕关节和髋关节-踝关节的相对向量来表示手臂和腿部的末端效应器的当前位置。采取这样的近似处理基于以下假设：股骨：胫骨和肱骨：尺骨的比例在个体之间的差异非常小。越来越多的研究表明，人虽然在身高和体重上的差别巨大，但是骨骼的比例确相差不多。

而这样的处理也允许训练完毕后的 IK 解算器能够适应具有不同几何长度和几何形状的的角色，即无需得到待解决角色的肢体长度，也能预测出符合自然动作和流畅的姿态序列，这在以往的研究中是不存在的。

通过这些技巧，我们可以构建神经网络监督学习中X 和Y 之间的映射：

- **X**：向量 N_X 表示从分层骨骼结构和动作联合通道中帧序列中末端效应器的全局位置。

- **Y**：向量 N_Y 表示身体各个关节方向的旋转角度值（BVH 文件中角度的定义为欧拉角）。训练集和测试集中的该值直接从动作文件的相应通道进行复制。后续的操作中，为了提高神经网络的训练过程的表现，防止深层神经网络出现梯度爆炸、梯度消失等情况，对该值做了数学方法的处理（标准化），但不改变该值的原始定义和所表达的意义。

3.1.2 IK 解算器关键组件

Motion Predictor（**MP**）和 Posture Estimator（**PE**）MP 和 PE 是我们 IK 解算器的两个主要组成部分，我们使用大致相同的神经网络结构来构建这两个组件（需要指出，网络的输入输出是不同的）。该网络的隐藏层是递归神经网络（RNN），具有三层 LSTM（每层的大小为 512），输入层和输出层是全连接层。

MP 网络结构 图 3.2说明了 **MP** 组件所用的神经网络的网络结构，该网络的输入是过去 **K** 帧中的 **EE** 位置序列，输出是下一个 **K** 帧中的 **EE** 位置序列。具体实现时，输入的帧数为 3 帧。设当前帧为第 i 帧，输入为第 $i-10$ 帧、第 $i-5$ 帧和第 i 帧。输入层有 9 个节点，分别表示输入帧各自的相对向量。输出的帧数为 2 帧。设当前帧为第 i 帧，输出为第 $i+5$ 帧和第 $i+10$ 帧。输出层共 6 个节点，分别表示输出帧各自的相对向量。将输入的帧数和输出的帧数进行整合后是 **PE** 的输入。

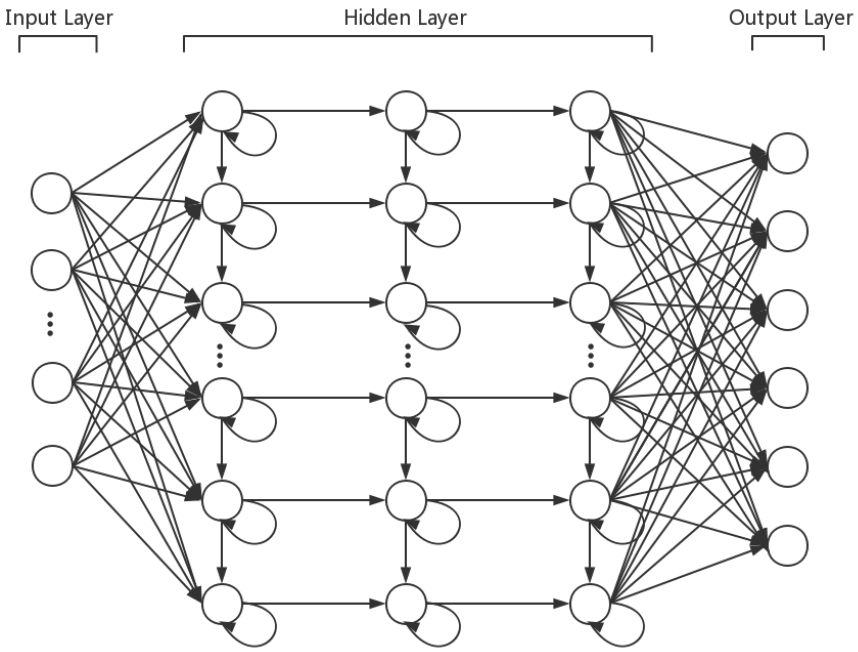


图 3.2: MP 组件的网络结构

MP 损失函数　该神经网络的损失函数定义为：

$$L_m = \sum_{i=0}^{K_m} (\mathbf{PE}_i - \mathbf{PE}_i^G) \tag{3.1}$$

该式中，K_m 表示连续帧的帧数，$\mathbf{PE}_i^G$ 表示来自运动捕捉数据库的真实值.

PE 网络结构　图 3.3说明了 **PE** 组件所用的神经网络的网络结构，该网络的输入是过去 **K** 帧中、当前帧和下一个 **K** 帧中的位置序列，输出是过去当前帧和下一个 **K** 帧中的 **EE** 预测姿势序列。具体实现时（以右手臂为例），输入的帧数为 5 帧。设当前帧为第 i 帧，输入的帧为第 $i-10$ 帧、第 $i-5$ 帧、第 i 帧和 **MP** 输出的第 $i+5$ 帧和第 $i+10$ 帧，即将 **MP** 的输入和输出进行整合。输入层有 15 个节点，分别表示输入帧的各自相对向量。输出的帧数为 6 帧。设当前帧为第 i 帧，输出的帧为第 i 帧、第 $i+1$ 帧、第 $i+2$ 帧、第 $i+3$ 帧、第 $i+4$ 帧、第 $i+5$ 帧。输出层有 36 个节点，，分别表示输出帧各自的肩关节和肘关节的旋转角。

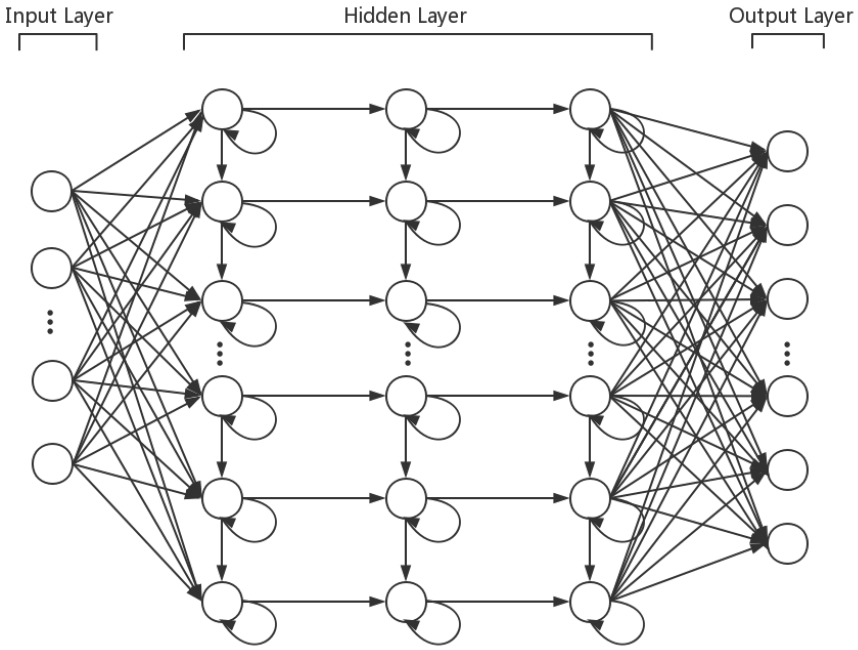


图 3.3：PE 组件的网络结构

PE 损失函数 神经网络的损失函数定义为：

$$L_p = \sum_{i=0}^{K_p} (\mathbf{P}_i - \mathbf{P}_i^G) \tag{3.2}$$

该式中，K_p 表示连续帧的帧数，$\mathbf{P}_i^G$ 表示来自运动捕捉数据库的真实值。

激活函数、优化器和超参数 深度神经网络采用目前流行的 RNN 网络模型作为网络连接模式以适用连续帧之间的时间关联性，具体细节如下：

- 选用线性整流函数（Rectified Linear Unit, ReLU）作为激活函数。线性整流函数定义了该神经元在线性变换 $\mathbf{w}^T\mathbf{x} + b$ 之后的非线性输出结果。更加直观的理解为，对于来自上一层神经网络的输入向量 x，该函数的神经元会输出 $\max\left(0, \mathbf{w}^T\mathbf{x} + b\right)$。下一层神经元以此作为本层神经网络的输入。

- 选用 Adam 优化器作为优化函数。Adam 结合 AdaGrad 和 RMSProp 两种优化算法

的优点，对梯度的一阶矩估计（First Moment Estimation，即梯度的均值）和二阶矩估计（Second Moment Estimation，即梯度的未中心化的方差）进行综合考虑，计算出更新步长。更新规则如下：

  - 计算时间梯度：$g_t = \nabla_\theta J\left(\theta_{\mathrm{t}-1}\right)$。
  - 计算指数式的梯度移动的平均数，β_1 为系数为指数衰减率，控制权重分配：$m_t = \beta_1 m_{t-1} + \left(1-\beta_1\right) g_t$；$m_0$ 初始化为 0。
  - 计算梯度平方的指数移动平均数，v_0 初始化为 0。β_2 系数为指数衰减率：$v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^2$。
  - 对梯度均值 m_t 进行偏差纠正,降低偏差对训练初期的影响:$\hat{m}_t = m_t / \left(1-\beta_1^t\right)$。
  - 对 v_t 进行纠正:$\hat{v}_t = v_t / \left(1-\beta_2^t\right)$。
  - 更新参数，初始的学习率 x 乘以梯度均值与梯度方差的平方根之比：$\theta_{\mathrm{t}} = \theta_{t-1} - \alpha^* \hat{m}_t / \left(\sqrt{\hat{v}_t} + \varepsilon\right)$。

- 设置 $batchsize = 128$，$learning_rate = 0.0001$，$maximalepoch = 1000$。

网络参数调节　在实际的训练过程中，为达到最好的训练效果，在模型的参数上和样本集的处理可能会做一些微调以达到更好的精度和更快的训练速度：

- 学习率自下降（learning rate adaptive decay）防止学习率过大，在收敛到全局最优点的时候会来回摆荡，所以学习率随着训练轮数不断按指数级下降，收敛梯度下降的学习步长。

- 梯度剪切（weithts regularization）训练集的规模的约束导致了神经网络的层数在设计过程中需要比过往经验中的更多。正因如此，梯度爆炸和梯度消失的现象有可能在网络模型的参数调整过程中出现。为了解决这一问题，设计一个梯度阈值，如果超过该阈值，直接将梯度置为该值。

3.1.3　去噪滤波器

平滑度 合成动作的平滑度是 IK 技术成功与否的关键衡量标准之一[16]，特别是在一些涉及收敛的迭代算法中。通常，建议程序在固定的迭代次数之后终止循环以避免程序陷入无限循环。然而，这个操作可能只能得到次优结果并导致合成动作的波动。

自然度 评估一个动作序列是否是自然的是困难的因为它没有一个固定的标准，除此之外，这个标准可能根据不同的场景和主题而随之变化。本次毕业论文中，在自然度的评估是通过比较合成动作和原始动作数据之间的差异，差异越小则代表动作越自然。之所以能做出这样的标准是因为本文采用的动作数据来自真实的人体动作捕捉，并非通过计算机模拟产生。**PE** 网络的输出存在一定的噪音（连续帧之间动作不够平滑和自然，更直观的说法为视觉上骨骼轻微抖动）进一步使用平均滤波器进行处理以进行去噪。平均滤波器的处理过程如下：

$$\mathbf{P} = \frac{1}{N}(\sum_{i}^{i+k} \mathbf{P}_i + \sum_{i-k}^{i} \mathbf{P}_j) \tag{3.3}$$

$\mathbf{P}_i$ 表示 **PE** 网络输出的连续预测姿势序列，$\mathbf{P_j}$ 表示由内存中读取的过往姿势序列。

具体实现中，设当前帧为第 x 帧，3.3中 $i = x-5$,$k = x+5$。从第 $x-5$ 帧至第 $x-1$ 帧为已知，因为是过往姿势序列，可以从内存中直接读取；从第 x 帧到第 $x+5$ 帧为 **PE** 网络的输出。最终得到的 **P** 为降噪后的姿势形态。这一步显著提高了 **PE** 输出预测姿势序列的平滑度，具体降噪效果见4.2.2。

3.2 姿势合成和姿势估计

本次毕业论文的 IK 求解器应用于两个应用：姿势合成和姿态估计。

3.2.1 姿势合成

在用户指定末端执行器的轨迹后，即末端执行器在三维世界坐标内的位置序列，以 0.5 秒的均匀间隔对轨迹进行采样，然后运用 **IK** 解算器自动生成全身姿势。轨迹的生成由两个步骤组成：相对坐标的计算和运动帧级联。相对坐标的计算指的是（以右手臂为例）：捕捉右臂的肩关节、肘关节、腕关节的末端执行器的位置，以 4.3.5中所示方法计算出某一个时刻某一个姿势唯一的右臂向量；运动帧级联指的是在构造输入数据时，加

入本文提出的连续帧之间的时间关联性，整合出符合训练好的解算器组件输入格式的数据。预测程序的流程如图 3.1所示。

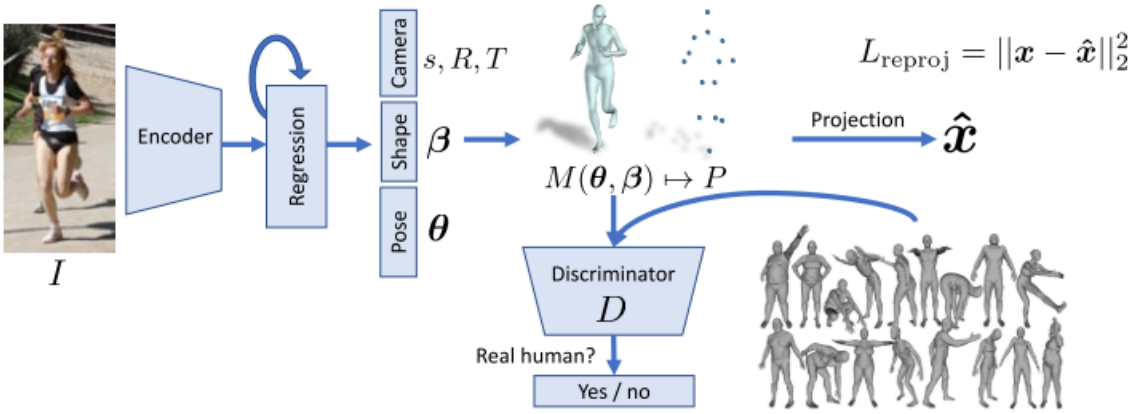


图 3.4: 2D 图像到 3D 人体骨骼预测流程[43]。输入的 2D 图像 I 首先通过 Encoder。Encoder 输出的结果传送到迭代 3Dregression 模块（该模块能够推断人的隐式三维表达方式来减少人体关节投影的误差）。3D 参数也被发送到 Discriminator，鉴别器（Discriminator）D 的目的是告知这些参数是否来自真实的人体形状和姿势。

3.2.2　姿态估计

本次毕业论文着重于改进使用单目摄像机捕捉图像到 3D 人体骨骼姿态的估计结果[43]。由于这是一个典型的不适定问题，所以解决该问题时具有一定的挑战性。

2D 图像到 3D 人体骨骼的预测大致框架如图 3.4所示。涉及论文中提出直接利用以人为中心的单张 RGB 图像来重建人体完整的三维网格模型。神经网络训练过程中，假设所有图像都使用真实的二维关节进行注释，该方法还考虑了一些具有三维注释的情况。

除此之外，该方法假设有一个具有不同形状和姿势的人体三维网格池。由于这些网格不一定有相应的图像与之对应，因此我们将此数据称为未配对数据。图 3.4展示了所提出的预测网络架构。该网络可以执行端到端的训练。为了推断 3D 人体网格和相机，一张图像的卷积特征被发送到迭代 3D 回归模块，使得其 3D 关节投射到带注释的 2D 关节上。预测的人体参数传送到一个鉴别器网络。鉴别器的任务是确定 3D 参数是否是来自未配对数据。这步操作可以促进网络输出基于多样式人体形态的 3D 人体网格，并且能够作为没有真实 3D 注释的原始图像的弱监督。得益于 $3L = \lambda(L_{\text{reproj}} + \text{⊬}L_{3\text{D}}) + L_{\text{adv}}\text{D}$

网格模型具有多种表达方式，数据驱动先验方法可以捕获关节角度限制，人体测量约束（例如身高，体重，骨骼比率）。当拥有真实 3D 信息时，将其作为中间损失。总的来说，该方法的目标

$$L = \lambda \left(L_{\text{reproj}} + \mathbb{1} L_{3D} \right) + L_{\text{adv}} \tag{3.4}$$

其中 控制每个物体的关联性的重要程度；$\mathbb{1}$ 是指示器函数，如果拥有真实值可用于图像则为 1，否则为 0。每个组件的具体实现如下：

- **3D 人体表达方式**该方法使用 **SMPL** 模型来编码人体的 3D 网格。形状参数 $\boldsymbol{\beta} \in \mathbb{R}^{10}$ 由 **PCA** 形状空间的前 10 个系数参数化得到。姿势参数 $\boldsymbol{\theta} \in \mathbb{R}^{3K} K = 23$ 个关节的相对 3D 旋转角来建模。**SMPL** 是一个可微函数，它输出一个有 $N = 6980$ 个顶点和 $M(\boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^{3 \times N}$ 的三角网格。通过线性回归得到的 3D 关键点（$X(\boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^{3 \times P}$）可以用于计算投影误差。

- **带回馈 3D 交互回归** 3D 回归模块的目标是输出带图像编码 ϕ 的 Θ 使得联合重投影误差 $L_{\text{reproj}} = \Sigma_i \left\| v_i \left(\mathbf{x}_i - \hat{\mathbf{x}}_i \right) \right\|_1$ 最小化。式中 $\mathbf{x}_i \in \mathbb{R}^{2 \times K}$ 是第 i 个真实 2D 关节。$v_i \in \{0, 1\}^K$ 是 K 个关节的各自是否可视化值。

  3D 回归模块将图像特征 θ 和当前参数 Θ_t 作为输入并输出残差 $\Delta\Theta_t$。通过将该残差加到当前估计 $\Theta_{t+1} = \Theta_t + \Delta\Theta_t$ 来更新参数。初始估计 Θ_0 被设置为平均 $\overline{\Theta}$。以下是 3D 损失的定义：

$$\begin{aligned} L_{3D} &= L_{3D\ \text{joints}} + L_{3D\ \text{smpl}} \\ L_{\text{joints}} &= \left\| \left(\mathbf{X_i} - \hat{\mathbf{X}}_\mathbf{i} \right) \right\|_2^2 \\ L_{\text{smpl}} &= \left\| [\boldsymbol{\beta}_i, \boldsymbol{\theta}_i] - \left[\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\theta}}_i \right] \right\|_2^2 \end{aligned} \tag{3.5}$$

- **分解对抗性先验**该工作使用经过训练的鉴别器网络 D 来判断 **SMPL** 参数是否对应于人体真实值。将此称为对抗性先验，因为鉴别器充当引导 3D 预测的数据驱动先验。该方法镜像 **SMPL** 的形状并进行了姿势分解，独立训练形状和姿势的鉴别器。因为姿势的预测基于动作树，所以进一步分解姿势鉴别器并单独为每个关节训练一个。为了捕获整个动作树的联合分布，训练一个接受所有旋转的鉴别器。由于每个鉴别器的输入是非常低的维度（β 为 $10-D$，每个关节为 $9-D$，所有关节

为 $9K-D$），因此训练的网络都可以是小型网络。所有姿势鉴别器共享旋转矩阵的共同特征空间，并且仅分别学习最终分类器。

在所有训练的 $k+2$ 个鉴别器中，每个鉴别器 D_i 的输出介于 $[0,1]$ 之间，用此来表示 Θ 来自数据库的可能性。实际训练过程中，使用最小二乘法来保证其稳定性。设 E 表示包括图像编码器和 3D 模块的编码器。编码器的对抗性损失函数是：

$$\min L_{\text{adv}}(E)=\sum \mathbb{E}_{\Theta\sim p_E}\left[(D_i(E(I))-1)^2\right] \square \tag{3.6}$$

每个鉴别器的目标是：

$$\min L(D_i)=\mathbb{E}_{\Theta\sim p_{\text{data}}}\left[(D_i(\Theta)-1)^2\right]+\mathbb{E}_{\Theta\sim p_E}\left[D_i(E(I))^2\right] \tag{3.7}$$

在实际的应用中，对于每个人体几何信息进行采集几乎是不可能的，所以本文所涉及的工作中，仅使用由 2D 图像到 3D 骨骼预测流程中输出的 **EE** 的 3D 位置和与其对应的根关节位置，并应用本文提出的 IK 解算器对当前身体骨骼连接做出一个最佳姿态的估计。

第四章　实验结果和讨论

4.1　实验细节

深度学习网络的训练环境和测试环境如下：

- 操作系统：Ubuntu 16.04
- CPU：Inter i7 7800k
- GPU：Nvidia Geforce TitanXP
- 内存：16G

实验所需的材料（包括代码、训练集、测试集）发布在 github 上[①]。整个训练过程需要大约 36 小时。

4.2　实验结果

4.2.1　训练集和测试集结果

神经网络的训练集和测试集输出结果全部通过 MATLAB 脚本替代原有的 BVH 文件的数据值，并通过 BVH PLAYER 进行播放。训练结果可以通过两方面来检测，一是 BVH 文件的播放后的视觉效果，二是观察训练过程中训练集和测试集的损失函数值。

图 4.2展示了来自训练集的合成动作（包括篮球运球、投篮和芭蕾舞动作）。这些示例都需要角色对象进行精细的操作（例如对篮球的掌控）以及与环境频繁的交互（例如地面上的脚部支撑）。这些动作的求解都需要 ik 解算器具备较高的精度。一般来说，训练样本的最优的平均损失 $< 10^{-5}$，实际的预测的欧拉角和真实值的相差约在 ± 0.01 之间。

[①]link: https://github.com/uhomelee/DeepInverseKinematicsSolver

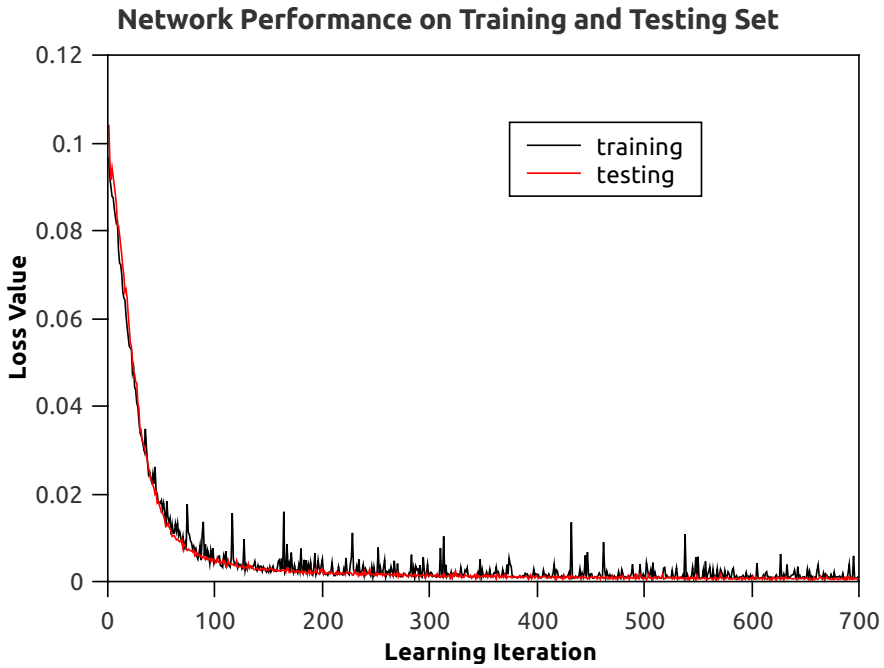
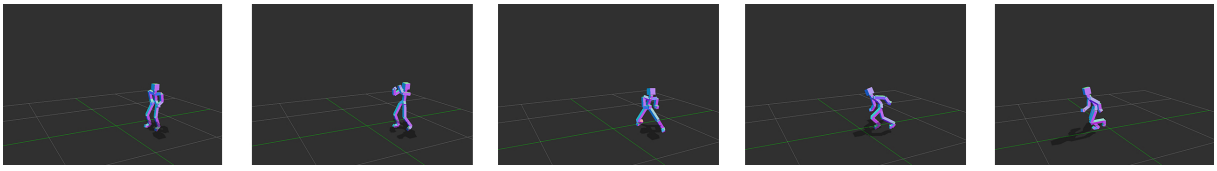
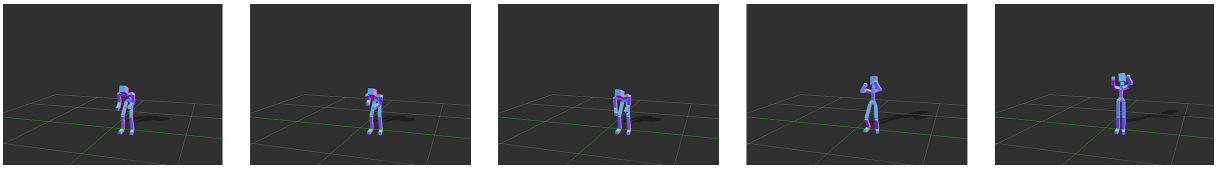


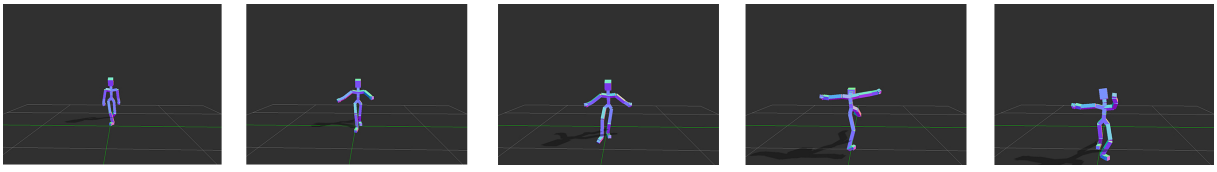
图 4.1: 训练集和测试集比较



(a) 篮球运球动作

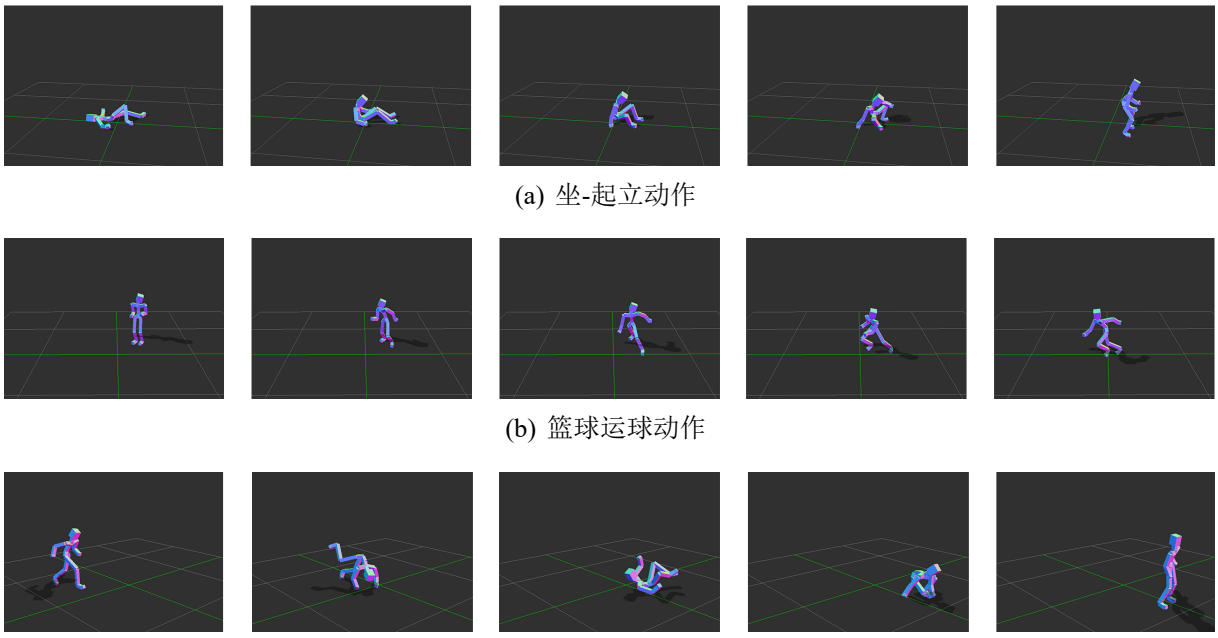


(b) 篮球投篮动作



(c) 芭蕾舞动作

图 4.2: 训练动作数据库合成动作



(a) 坐-起立动作

(b) 篮球运球动作

(c) 翻跟头动作

图 4.3: 测试动作数据库合成动作

图 4.3展示了来自测试集中的合成动作（包括从地面起立、篮球运球和翻跟斗）。尽管这些例子中的动作并未包含在训练集中，但 IK 解算器仍可以解决该问题并产生符合自然姿态的动作。需要指出的是，测试集上的结果并未出现过拟合的情况，如图 4.1所示，训练集和测试集的损失函数值随着迭代次数的递增而下降，最终的结果测试样本的平均损失在 10^{-4} 10^{-5} 之间。通过实验，在全连接（FCN）模型中，虽然训练集上的网络性能表现良好，但在测试集上非常容易出现过拟合的情况。

4.2.2 去噪滤波器结果

尽管网络模型的损失函数在训练结束后收敛得非常小，但 **IK** 问题的求解的另一个需要考虑的因素是合成连贯动作后的自然度。考虑到 **PE** 组件的输出在 BVH PLAYER 上播放时在某些复杂的动作（例如频繁的上下抬手、大范围的移动等情况）时出现的视觉上的抖动问题，对 **PE** 组件的输出进行了降噪，具体的方法见3.1.3。如图 4.4所示，降噪的结果可以说是非常显著的。在 BVH PLAYER 上播放后，原先已经符合视觉上自然流畅的动作会显得更加自然（避免了太大角度的弯曲，违背了人体关节的旋转极限）；原先有抖动的文件，在经过降噪后，基本可以做到流畅和自然。

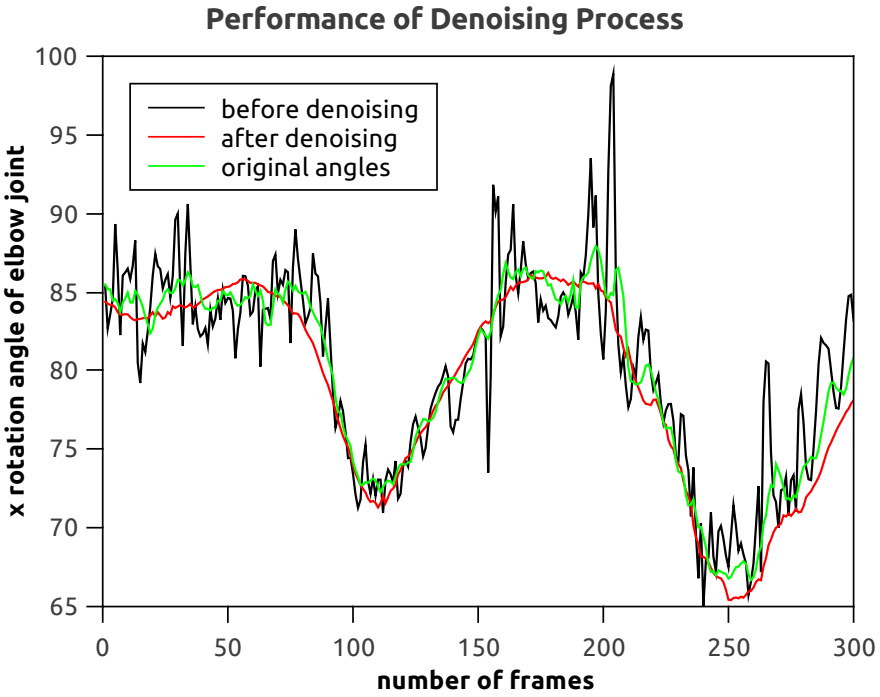
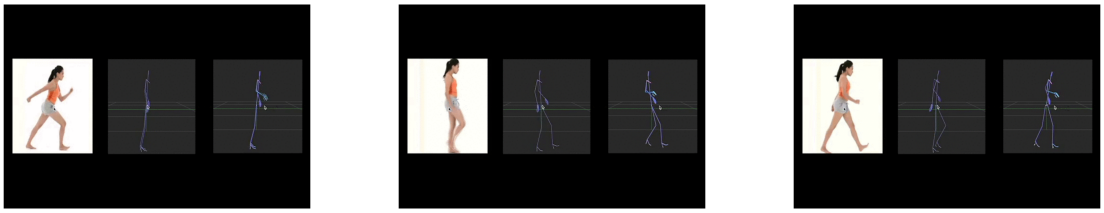


图 4.4: 去噪滤波器处理前后和真实值对比



(a)　(b)　(c)

图 4.5: 2D 图像到 BVH 文件转换

4.2.3　姿态估计结果

如图 4.5静态所示（左边为原始图片，中间为原始的姿态估计的结果[43], 右边为运用 **IK** 解算器修正后的图片）, 原始的姿态估计结果，有一些动作不符合人体骨骼活动的自然规律，例如肘关节向外翻转。但与此同时可以注意到，某一些关节，例如肩关节和腕关节，在姿态估计中的准确度远远高于肘关节这类活动范围更大的关节。所以将 **IK** 解算器姿态估计问题中时，以右手臂为例，只考虑肩关节和腕关节的三维坐标，运用解算器反向推倒出肘关节的旋转角，然后再将其转换为世界坐标。在动态的 GIF 图中可以观察到，运用 **IK** 解算器后的右手臂的弯曲更为自然。

除此之外，本文的去噪滤波器在姿态估计中也不可忽视。图 4.5中的姿态已经经过去噪滤波器降噪处理过。在动态的 GIF 中，原始的姿态估计结果的抖动非常明显，甚至 root 关节点的世界坐标都呈现出不规律的抖动；但在运用了本文的去噪滤波器方法后，整体的运动消除了抖动，呈现出平滑效果，符合人体运动规律。

4.3 讨论

4.3.1 不同网络结构的比较

图4.6验证了本文的网络选择的有效性，将其与其他流行网络进行性能比较，包括卷积神经网络（CNN），全连接神经网络（FCN）和生成性对抗网络（GAN）。这些网络的实现代码和模型在公开 Github 上（请参阅本文 4.1中的链接）

结果表明，具有相当小节点的 RNN 实现了与比其节点多的 FCN 相当的性能。因为 RNN 具有对时序序列表现更加优秀的特性，而本文中提出的对原始的数据经过额外的整合，加入了时间关联性，使之更符合 RNN 网络输入数据的特征要求。同时，FCN 比起 RNN 非常可能导致过拟合问题，虽然在训练集上表现相当，但在测试集上 FCN 的表现远远不如 RNN。在实际情况中，FCN 只学习训练集的数据，相当于记忆了一个庞大的数据库，测试集上的损失函数并未明显下降，从另一方面证明 FCN 网络并未学习到如何求解 **IK** 问题。

实验结果同样证明，CNN 和 GAN 在本文的案例中不能产生令人满意的表现。CNN 在实验过程，损失函数始终无法下降；GAN 的调节参数过于复杂，且该问题是一个典型的监督学习问题，GAN 的学习特征与之并不符合，所以 GAN 网络的损失函数一直无法收敛也可以得到解释。

4.3.2 不同帧数输入的比较

图4.7比较了当不同数量的帧用作网络输入时的学习性能。该图表明，当单个帧用作网络输入时，学习性能不会产生令人满意的结果。实验中，将输入一帧的训练后的模型经过反算得到的 BVH 文件播放后，呈现出了巨大的抖动，且不符合人体运动规律。当

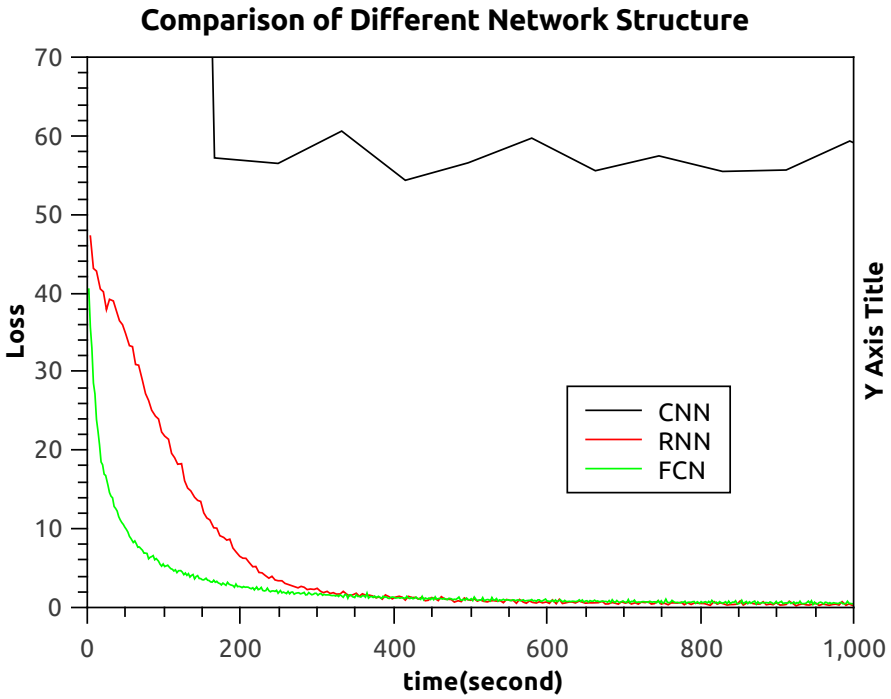


图 4.6：不同网络模式的学习性能比较

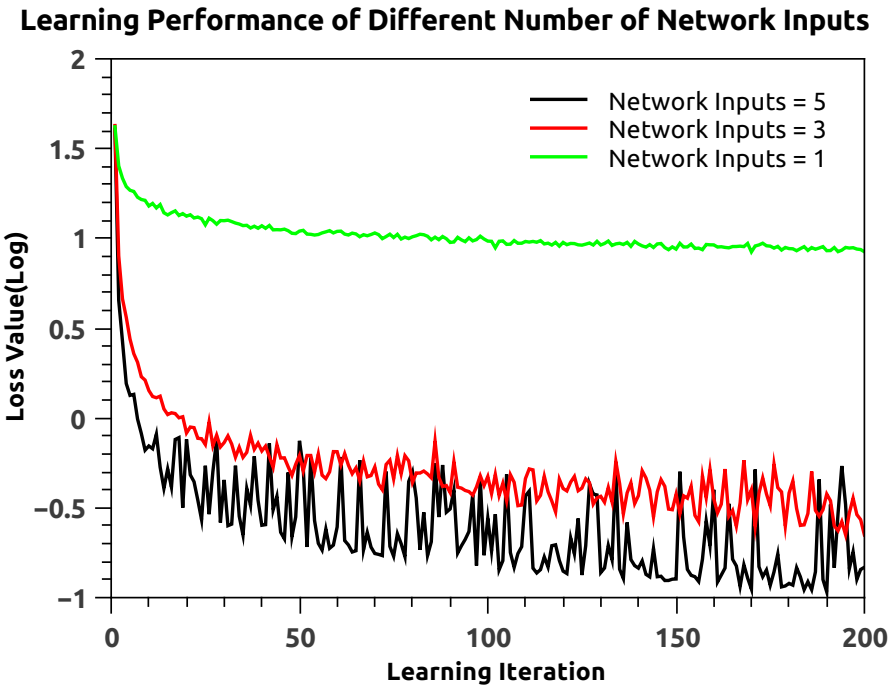


图 4.7：不同输入帧数输入的网络学习性能比较

增加帧数作为输入时，学习性能得到改善，但损失的方差同时也稍微更大。实验中，如图 4.8所示，试验了输入 7 帧以上的结果，并没有太大的提升效果。为了学习性能、内存空间和稳定性的平衡，本文选取网络输入为 5。

LOG（minimum loss after 500 literation）				
input / output	1	3	5	7
1	-1.02831213	-2.9655274	-3.50842988	-3.52685885
4	-1.07631704	-3.00338568	-3.47689099	-3.49698002
6	-1.07733244	-2.99461726	-3.51613597	-3.47351756

图 4.8: 不同输入帧数和输出帧数的网络学习性能比较

4.3.3 左臂和右臂的比较

图4.9比较了求解左侧手臂和右侧手臂时 IK 求解器的学习性能。两种情况下的网络学习表现相似，只是左臂的函数损失相对较小且更稳定（因为损失函数做了 log 变换，将原来的细节进行了放大，实际情况中，二者差别更小一些）。这种差异应该是由于右臂负责更复杂的运动造成的（除了以左手为惯用手的少数群体）。在 CMU 的 Mocap 数据库中，经过排查后，几乎没有左撇子的参与者（考察了篮球运动类似的需要区分左右手的情况）。与左臂相比，一个以右手为惯用手的人右臂往往负责具有更高的复杂性的动作，活动范围也更大。这是导致右臂的 **IK** 的 IK 解算器具有更高的函数损失和更多的抖动变化的主要原因。

4.3.4 手臂和腿的比较

图4.10比较了在求解右侧的腿部和手臂的 IK 问题时网络的学习性能。结果表明，脚的解算器的平均损失和方差远小于手臂的平均损失和方差。这反映了一个事实，手部运动比脚部运动更加细腻和复杂。根据人类常识，手进行的活动比腿部更为复杂，且数量更多，且由于韧带和人体结构的影响，手臂的活动范围远远大于腿部。

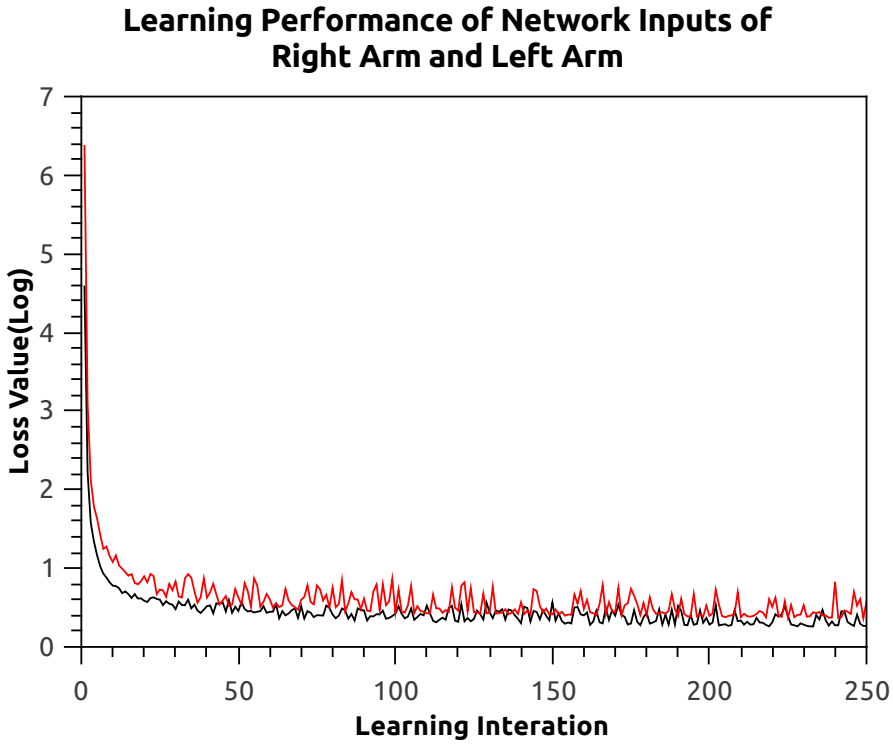


图 **4.9:** 左臂和右臂的网络学习性能表现

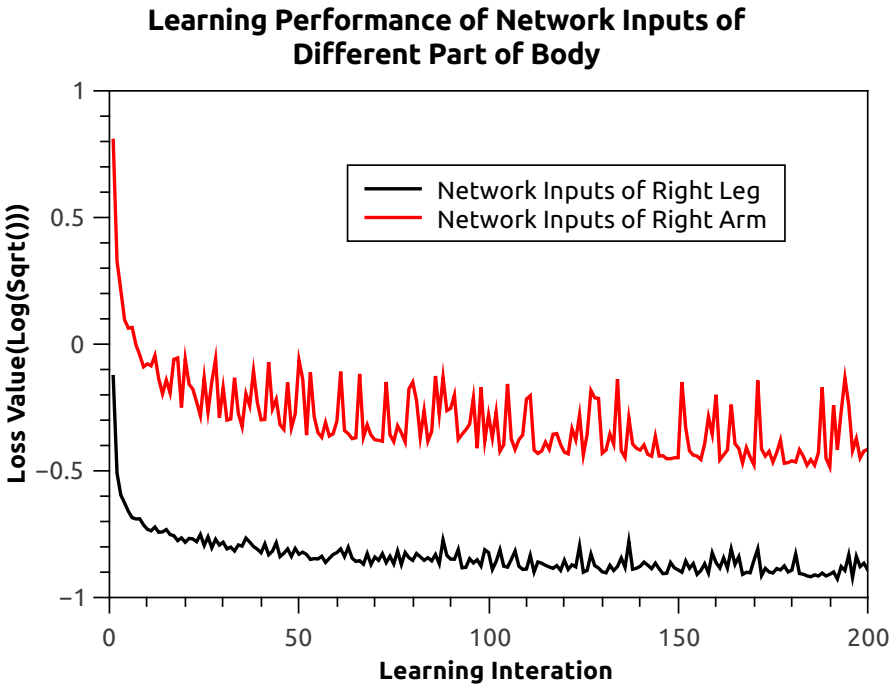
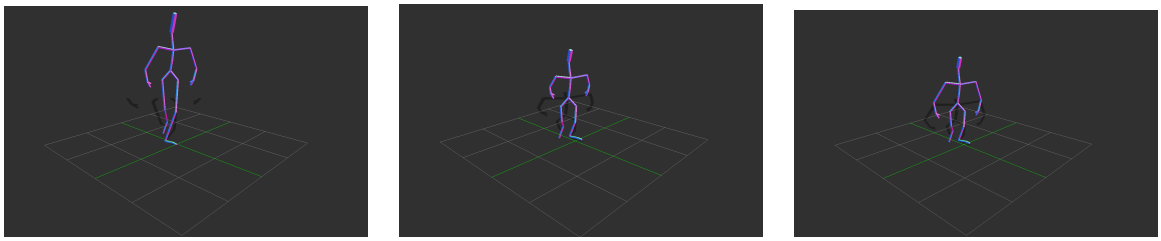


图 **4.10:** 手臂和腿部的网络学习性能比较



(a) 站立动作（正常肢体长度） (b) 站立动作（短臂） (c) 站立动作（短腿）

图 4.11: 不同肢体长度角色的合成动作

4.3.5 重定位动作至不同肢体长度角色

由于末端执行器的位置用肢长度进行了标准化，以右手臂为例，

$$\overrightarrow{AC} = \frac{(\overrightarrow{AB} + \overrightarrow{BC})}{(\left|\overrightarrow{AB}\right| + \left|\overrightarrow{BC}\right|)}$$

式中 $\overrightarrow{AB}$、$\overrightarrow{BC}$、$\overrightarrow{AC}$ 皆为三维向量，分别表示右大臂、右小臂和 IK 解算器输入的向量.**A** 表示肩关节的位置，**B** 表示肘关节位置，**C** 表示腕关节位置. 图 4.11展示了本文方法可以推广到具有不同肢长的角色的 IK 问题的求解. 三张图内虽然肢体长度差别巨大，但在 BVH PLAYER 中进行播放时，手臂和腿部的活动并未出现滑动和抖动，皆符合人体运动的规律。

4.4 应用

4.4.1 关键帧轨迹

本文涉及工作的一个直接应用就是关键帧轨迹。用户将末端效应器的运动轨迹拟合为一条样条曲线，IK 解算器可以自动生成沿着身体关节连接方向的各个关节角度，以获得四肢完整的关节运动轨迹。该任务首先应使用均匀参数对样条曲线进行采样，其次还需要将连续多帧合并为一段输入作为神经网络的输入。

4.4.2　实时动作压缩

本文方法的另一个应用是动作压缩。在标准类型的动作文件（例如 BVH 格式）中，角色动作用根关节的全局变换和子关节相对于父关节的偏移和方向来描述。鉴于 IK 求解器具有高精度，因此用末端执行器的位置替换关节的角度和方向是足够的。这种方法可以将动作的存储空间减少至少 50%，这对于实时应用程序中的数据传输非常的重要。

4.4.3　频繁交互环境中的运动合成

图 4.2和图 4.3均展示了在涉及频繁的肢体与环境接触的情景，本文所述的方法仍具有合成自然的动作的能力。

第五章　总结

本文提出了一种利用深度学习神经网络求解 IK 问题的方法，特别是目前涉及此问题最为广泛的机械臂领域。此 IK 解算器能够处理涉及复杂人体-环境交互的场景，与此同时可以生成与真实动作最相似的骨骼姿态。本次毕业论文的主要贡献可以概括如下：

- 将深度学习技术应用至该问题。不同于以往的其他论文中的工作，做到了大规模的数据集上的训练并得到了可靠的测试集结果。
- 定义了输入帧和输出帧之间的临界时间相关性，并将之与目前应用广泛的 LSTM 单元进行结合。
- 使用了恒定肢体比率的假设和相对坐标，消除了以往工作中的个体之间的差别性，可以真正做到无需给出肢体长度等先验信息的基础上进行 IK 问题的求解。
- 对 IK 问题的求解的正确性进行了定义，并对 IK 解算器两个组件的输出进行了降噪，且降噪结果明显。
- 对基于单目摄像机的 3D 人体骨骼工作进行了复现，并运用 IK 解算器和去噪滤波器对结果进行了改进。

对于未来的工作，可以分为以下 X 点：

- 对四肢的 IK 解算器建立关联性（如利用树网络），做到全身的 end-to-end 的训练。
- 将其他信息（如图片语义信息）等加入网络训练中，以此来改善网络的求解自然度。
- 将本文的 IK 解算器应用于全身运动捕捉，并在末端执行器上安装惯性传感器，以此来改善合成动作的自然性。
- 继续其他网络的求解性，如利用强化学习来解决数据库规模问题等。

参考文献

[1] REINHART R F, STEIL J J. Neural learning and dynamical selection of redundant solutions for inverse kinematic control[C]//Ieee-Ras International Conference on Humanoid Robots. [S.l.: s.n.], 2011: 564–569.

[2] GROCHOW K, MARTIN S, HERTZMANN A, et al. Style-based inverse kinematics[J]. Siggraph Acm Siggraph Papers, 2004, 23(3): 522–531.

[3] TOURNIER M, WU X, COURTY N, et al. Motion compression using principal geodesics analysis[C]//Computer Graphics Forum: volume 28. [S.l.]: Wiley Online Library, 2009: 355–364.

[4] HUANG J, WANG Q, FRATARCANGELI M, et al. Multi-variate gaussian-based inverse kinematics[C]//Computer Graphics Forum: volume 36. [S.l.]: Wiley Online Library, 2017: 418–428.

[5] KöKER R, ÖZ C, ÇAKAR T, et al. A study of neural network based inverse kinematics solution for a three-joint robot[J]. Robotics and Autonomous Systems, 2004, 49(3): 227–234.

[6] DAYA B, KHAWANDI S, AKOUM M. Applying neural network architecture for inverse kinematics problem in robotics[J]. Journal of Software Engineering and Applications, 2010, 3(03): 230.

[7] FENG Y, YAO-NAN W, YI-MIN Y. Inverse kinematics solution for robot manipulator based on neural network under joint subspace[J]. International Journal of Computers Communications and Control, 2012, 7(3): 459–472.

[8] LEVINE S, KOLTUN V. Learning complex neural network policies with trajectory optimization[C]//International Conference on Machine Learning. [S.l.: s.n.], 2014: 829–837.

[9] HOLDEN D, SAITO J, KOMURA T. A deep learning framework for character motion synthesis and editing[J]. ACM Transactions on Graphics (TOG), 2016, 35(4): 138.

[10] HOLDEN D, KOMURA T, SAITO J. Phase-functioned neural networks for character control[J]. ACM Transactions on Graphics (TOG), 2017, 36(4): 42.

[11] PIETAK A, MA S, BECK C W, et al. Fundamental ratios and logarithmic periodicity in human limb bones[J]. Journal of anatomy, 2013, 222(5): 526–537.

[12] CHOSET H, LYNCH K, HUTCHINSON S, et al. Principles of robot motion: Theory, algorithms, and implementation[Z]. [S.l.: s.n.], 2005.

[13] BROGAN D, METOYER R, HODGINS J. Dynamically simulated characters in virtual environments[Z]. [S.l.: s.n.], 1998.

[14] COURTY N, ARNAUD E. Inverse kinematics using sequential monte carlo methods[C]// International Conference on Articulated Motion and Deformable Objects. [S.l.]: Springer, 2008: 1–10.

[15] HECKER C, RAABE B, ENSLOW R W, et al. Real-time motion retargeting to highly varied user-created morphologies[C]//ACM Transactions on Graphics (TOG): volume 27. [S.l.]: ACM, 2008: 27.

[16] ARISTIDOU A, LASENBY J, CHRYSANTHOU Y, et al. Inverse kinematics techniques in computer graphics: A survey[C]//Computer Graphics Forum: volume 37. [S.l.]: Wiley Online Library, 2018: 35–58.

[17] LEE C. Dynamically simulated characters in virtual environments[Z]. [S.l.: s.n.], 1984.

[18] TOLANI D, GOSWAMI A, I N. Badler real-time inverse kinematics techniques for anthropomorphic limbs[Z]. [S.l.: s.n.].

[19] BUSS S R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods[J]. IEEE Journal of Robotics and Automation, 2004, 17 (1-19): 16.

[20] UNZUETA L, PEINADO M, BOULIC R, et al. Full-body performance animation with sequential inverse kinematics[J]. Graphical models, 2008, 70(5): 87–104.

[21] MUKHERJEE S, PARAMKUSAM D, DWIVEDY S K. Inverse kinematics of a nao humanoid robot using kinect to track and imitate human motion[C]//Robotics, Automation, Control and Embedded Systems (RACE), 2015 International Conference on. [S.l.]: IEEE, 2015: 1–7.

[22] HARISH P, MAHMUDI M, CALLENNEC B L, et al. Parallel inverse kinematics for multithreaded architectures[J]. ACM Transactions on Graphics (TOG), 2016, 35(2): 19.

[23] COLOMÉ A, TORRAS C. Redundant inverse kinematics: Experimental comparative review and two enhancements[C]//Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. [S.l.]: IEEE, 2012: 5333–5340.

[24] ZHAO J, BADLER N I. Inverse kinematics positioning using nonlinear programming for highly articulated figures[J]. ACM Transactions on Graphics (TOG), 1994, 13(4): 313–336.

[25] ROSE C, GUENTER B, BODENHEIMER B, et al. Efficient generation of motion transitions using spacetime constraints[C]//Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. [S.l.]: ACM, 1996: 147–154.

[26] KENWRIGHT B. Inverse kinematics–cyclic coordinate descent (ccd)[J]. Journal of Graphics Tools, 2012, 16(4): 177–217.

[27] ARISTIDOU A, LASENBY J. Fabrik: A fast, iterative solver for the inverse kinematics problem[J]. Graphical Models, 2011, 73(5): 243–260.

[28] ROLF M, STEIL J J, GIENGER M. Goal babbling permits direct learning of inverse kinematics[J]. IEEE Transactions on Autonomous Mental Development, 2010, 2(3): 216–229.

[29] ROSE III C F, SLOAN P P J, COHEN M F. Artist-directed inverse-kinematics using radial basis function interpolation[C]//Computer Graphics Forum: volume 20. [S.l.]: Wiley Online Library, 2001: 239–250.

[30] CHAI J, HODGINS J K. Performance animation from low-dimensional control signals [C]//ACM Transactions on Graphics (ToG): volume 24. [S.l.]: ACM, 2005: 686–696.

[31] CARVALHO S R, BOULIC R, THALMANN D. Interactive low-dimensional human motion synthesis by combining motion models and pik[J]. Computer Animation and Virtual Worlds, 2007, 18(4-5): 493–503.

[32] RAUNHARDT D, BOULIC R. Motion constraint[J]. The Visual Computer, 2009, 25 (5-7): 509.

[33] PETRU MAIOR" UNIVERSITY OF TG. MUREş T R, No. 1 N.Iorga St. Neural network based inverse kinematics solution for trajectory tracking of a robotic arm[Z]. [S.l.: s.n.].

[34] REINHART R F, STEIL J J. Neural learning and dynamical selection of redundant solutions for inverse kinematic control[Z]. [S.l.: s.n.].

[35] LUKOSEVICIUS M, JAEGER H. Reservoir computing approaches to recurrent neural network training[J]. Computer Science Review, 2009.

[36] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: a new learning scheme of feed forward neural networks[J]. IEEE Intern. Joint Conf. on Neural Networks.

[37] SIDENBLADH H, BLACK M J, SIGAL L. Implicit probabilistic models of human motion for synthesis and tracking[J]. Proc. ECCV, LNCS 2353, vol. 1, 784−800, 2002.

[38] CSISZAR A, EILERS J, VERL A. On solving the inverse kinematics problem using neural networks[C]//Mechatronics and Machine Vision in Practice (M2VIP), 2017 24th International Conference on. [S.l.]: IEEE, 2017: 1–6.

[39] Wikipedia contributors. Artificial neural network — Wikipedia, the free encyclopedia [EB/OL]. 2019. https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=893208691.

[40] Wikipedia contributors. Deep learning — Wikipedia, the free encyclopedia[EB/OL]. 2019. https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=894592567.

[41] 维基百科. 递归神经网络 — 维基百科, 自由的百科全书[EB/OL]. 2019. https://zh.wikipedia.org/w/index.php?title=%E9%80%92%E5%BD%92%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&oldid=53141336.

[42] 维基百科. 長短期記憶 — 维基百科, 自由的百科全书[EB/OL]. 2019. https://zh.wikipedia.org/w/index.php?title=%E9%95%B7%E7%9F%AD%E6%9C%9F%E8%A8%98%E6%86%B6&oldid=53142858.

[43] KANAZAWA A, BLACK M J, JACOBS D W, et al. End-to-end recovery of human shape and pose[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2018: 7122–7131.

[44] CHOI B B, LAWRENCE C. Inverse kinematics problem in robotics using neural networks [J]. Nasa Sti/recon Technical Report N, 1992, 93.

[45] DUKA A V. Neural network based inverse kinematics solution for trajectory tracking of a robotic arm[J]. Procedia Technology, 2014, 12(1): 20–27.

[46] BAYO E, PAPADOPOULOS P, STUBBE J. Inverse dynamics and kinematics of multilink elastic robots: An iterative frequency domain approach[Z]. [S.l.: s.n.].

[47] LIN M T, LIN H B, LIU C C, et al. Algebraic-elimination based solution of inverse kinematics for a humanoid robot finger[J]. International Conference on Mechatronics and Automation, 2011.

[48] WITKIN A, POPOVIC′ Z. Motion warping[J]. Proceedings of SIGGRAPH 95 (Aug.), 1995.

[49] ROSE C M F B B, C. Verbs and adverbs: Multidimensional motion interpolation[J]. IEEE Computer Graphics Applications 18, 5, 32–40., 1998.

[50] REINHART R F, STEIL J J. Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot icub[C]//Ieee-Ras International Conference on Humanoid Robots, 2009. Humanoids. [S.l.: s.n.], 2010: 323–330.

[51] NEUMANN K, ROLF M, STEIL J J, et al. Learning inverse kinematics for pose-constraint bi-manual movements[Z]. [S.l.: s.n.], 2010.

[52] WAEGEMAN T, SCHRAUWEN B. Towards learning inverse kinematics with a neural network based tracking controller[C]//Neural Information Processing - International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings. [S.l.: s.n.], 2011: 441–448.

[53] ERLEBEN K, ANDREWS S. Solving inverse kinematics using exact hessian matrices[J]. Computers & Graphics, 2019, 78: 1–11.

[54] WU X, TOURNIER M, REVERET L. Natural character posing from a large motion database[J]. IEEE computer graphics and applications, 2011, 31(3): 69–77.

[55] HO E S, SHUM H P, CHEUNG Y M, et al. Topology aware data-driven inverse kinematics [C]//Computer Graphics Forum: volume 32. [S.l.]: Wiley Online Library, 2013: 61–70.

[56] SHUM H P, HO E S, JIANG Y, et al. Real-time posture reconstruction for microsoft kinect. [J]. IEEE Trans. Cybernetics, 2013, 43(5): 1357–1369.

[57] LV P, XU M, YANG B, et al. Data-driven humanlike reaching behaviors synthesis[J]. Neurocomputing, 2016, 177: 26–32.

[58] CASERMAN P, GARCIA-AGUNDEZ A, KONRAD R, et al. Real-time body tracking in virtual reality using a vive tracker[J]. Virtual Reality, 2018: 1–14.

[59] CHAI J, HODGINS J K. Constraint-based motion optimization using a statistical dynamic model[J]. ACM Transactions on Graphics (TOG), 2007, 26(3): 8.

[60] RAGHAVAN M, ROTH B. Inverse kinematics of the general 6r manipulator and related linkages[J]. Journal of Mechanical Design, 1993, 115(3): 502–508.

[61] MANOCHA D, CANNY J F. Efficient inverse kinematics for general 6r manipulators[J]. IEEE transactions on robotics and automation, 1994, 10(5): 648–657.

[62] GAN J Q, OYAMA E, ROSALES E M, et al. A complete analytical solution to the inverse kinematics of the pioneer 2 robotic arm[J]. Robotica, 2005, 23(1): 123–129.