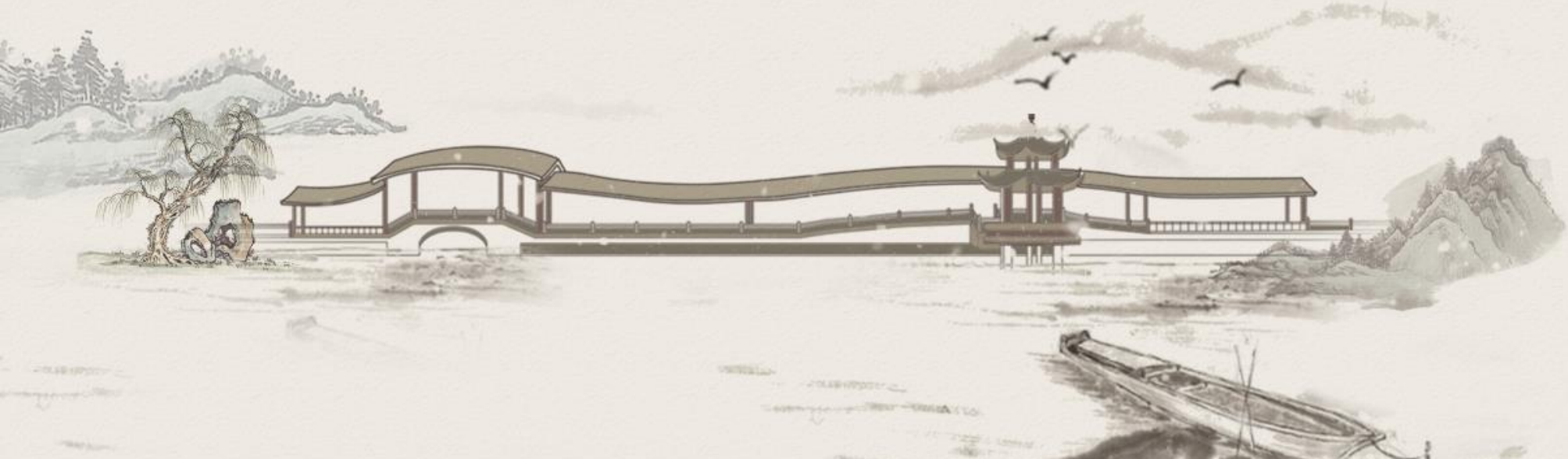
Predicting New Workload or CPU Performance by Analyzing Public Datasets

ACM Transactions on Architecture and Code Optimization

通过分析公共数据集来预测新的工作负载或CPU性能

汇报人：胡由钻

目录



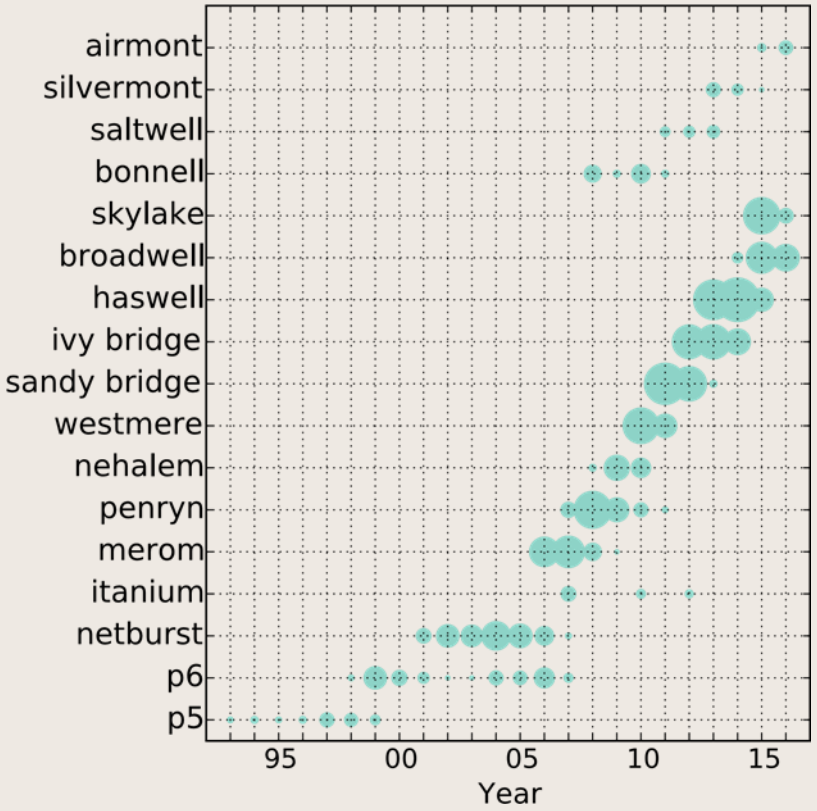
01 问题与背景

Intel CPU的架构

微架构又称为微处理器体系结构。是在计算机工程中，将一种给定的指令集架构在处理器中执行的方法。一种给定指令集可以在不同的微架构中执行。计算机架构是微架构和指令集设计的结合。

自1989年起英特尔就一直遵循着其称为“Tick-Tock模式”的新产品创新节奏，即每两年交替推出新一代的先进制程技术和处理器微体系架构。2016年之后，英特尔的Tick-Tock模式转换为“制程框架优化”三要素模式。

SKU: Stock Keeping Unit，一般指生产商给自家产品的编号。SKU具有不同的特性，如频率、缓存大小、内存带宽和核心计数。本文中一个SKU代表一个处理器号



英特尔在1993年至2016年间为17个微架构发布的SKU数量。气泡大小表示SKU的数量

CPU性能

- CPU的性能取决于

微处理器的性能不仅仅是它的微体系结构的功能的体现；它关键地取决于运行在它上面的工作负载的性质。因此，在量化实际的CPU性能时，需要同时考虑CPU微体系结构和工作负载。可以给消费者理性的参考

- 测试方法——基准测试

1996 International Workshop on Structural Control 会议倡导建立Benchmark结构 。基准测试就是采用同一的标准规范进行测试。目前比较流行的基准测试套件有：Geekbench、SPEC CPU和Passmark。它们收集了大量的性能数据存储库，并允许公众比较已知配置在标准工作负载上的性能。

- 存在的问题

首先，消费者需要等待新处理器的全面基准测试结果提供给公共存储库。其次，对于新的工作负载，消费者必须测试大量可能的配置，才能找到最有效的硬件。由于存在着这两个问题，一些消费者盲目地依赖基准库中的CPU总分数

- 本文的解决办法

为了解决这些问题，我们使用统计方法和机器学习技术来分析SPEC CPU2006和Geekbench 3存储库中的数据。SPEC CPU2006在学术研究中被广泛使用，Geekbench 3被许多消费者用来做购买决策。我们使用深度神经网络（DNNs）来预测英特尔CPU的性能，并将DNN预测与线性回归（LR）进行比较。DNN预测模型学习不同工作负载的处理器规范之间的相互关系，提高基准数据存储库的实用性。

CPU性能的度量

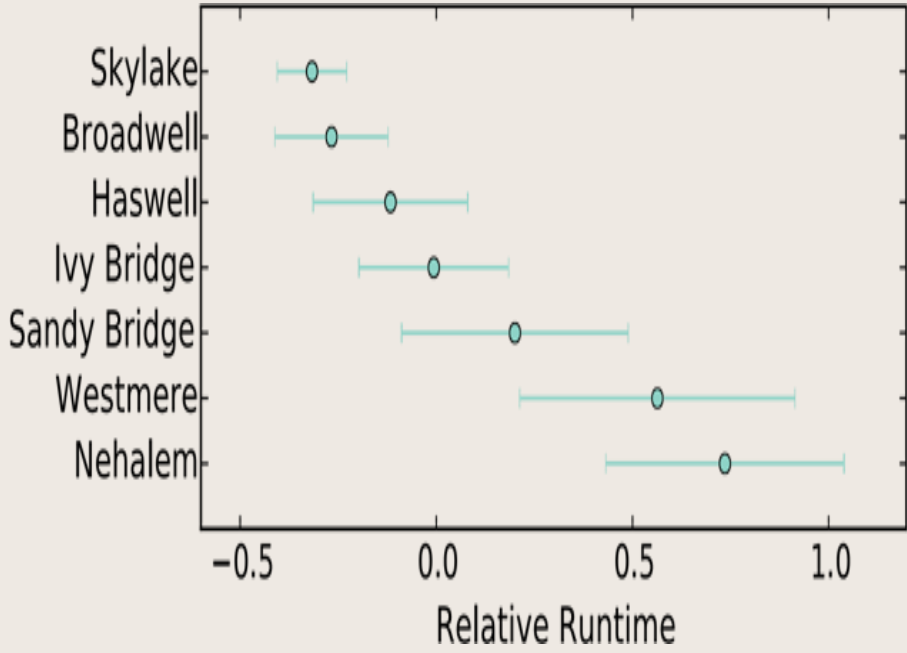
在本文中，我们使用相对运行时作为性能的度量。我们以Sandy Bridge处理器（E3-1230）为参考机器。工作负荷的相对运行时定义为：

$$relative\ runtime = \frac{runtime - runtime_{ref}}{runtime_{ref}}$$

其中$runtime_{ref}$是引用计算机上相应工作负载的运行时

E3-1230的相对运行时间为0在图中，E3-1230的性能略低于所有Sandy Bridge SKU的平均值，因此该行以大于0的值为中心。因为所有结果都是相对的对于相同的参考处理器，结果不受参考的特定选择的影响。

为了显示不同SKU的不同性能，我们采用SKU上的平均SPEC性能，并根据其微体系结构代码名称收集SKU。在图中，点显示具有相同微体系结构代码名称的SKU的平均相对运行时间，条形的长度显示标准偏差。更长的条形表明SKU的性能更加多样化。 SKU具有相同的微体系结构，但具有不同的频率，高速缓存大小，内存大小等。条形水平重叠。这意味着在选择SKU以优化性能时，可以有许多选择，具有不同的微体系结构但性能相似。

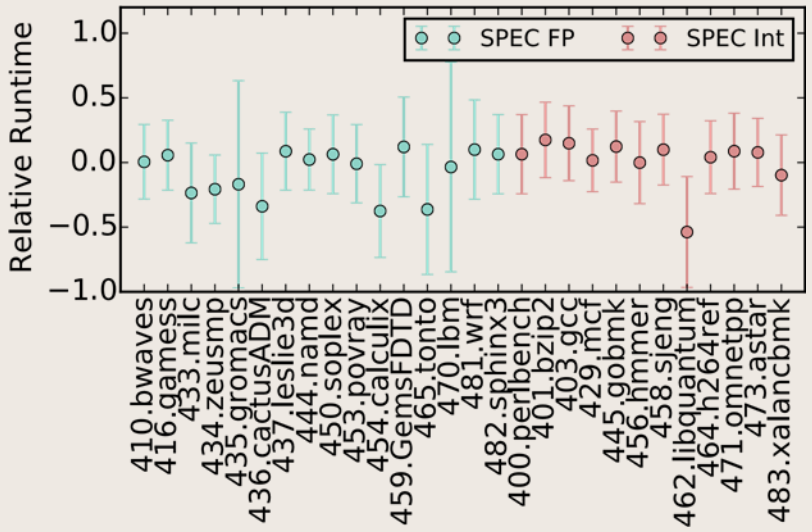


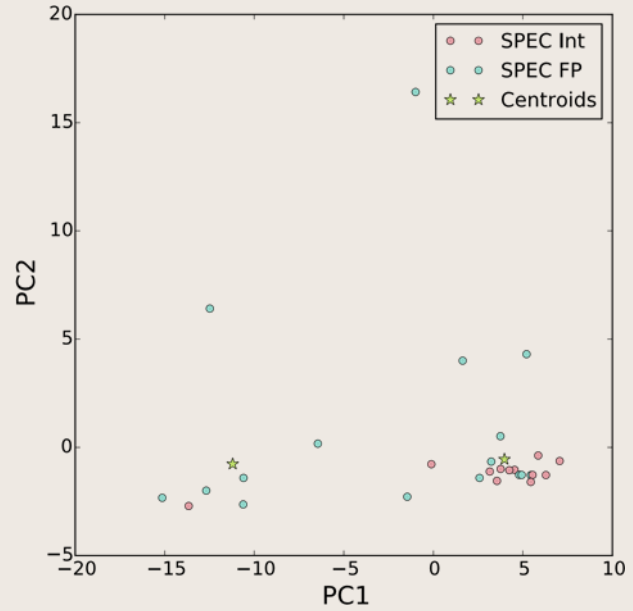
不同负载的影响

在SPEC不同负载的CPU平均运行时不同：负载分为SPEC FP与SPEC Int两类，测试了SPEC的352个SKU共计639个配置

352个SKU的639个配置进行规范工作负载性能扩展的主成分分析。绿点是两个由K-mean识别的类的质心。SPEC FP工作负载比SPEC Int分布得更广。462.libquantum（左下角的红点）与SPEC FP基准聚集在一起。靠近顶部的离群值是481.wrf。

数据集中SKU在SPEC工作负载相对运行时间的平均值和标准偏差





本文的工作

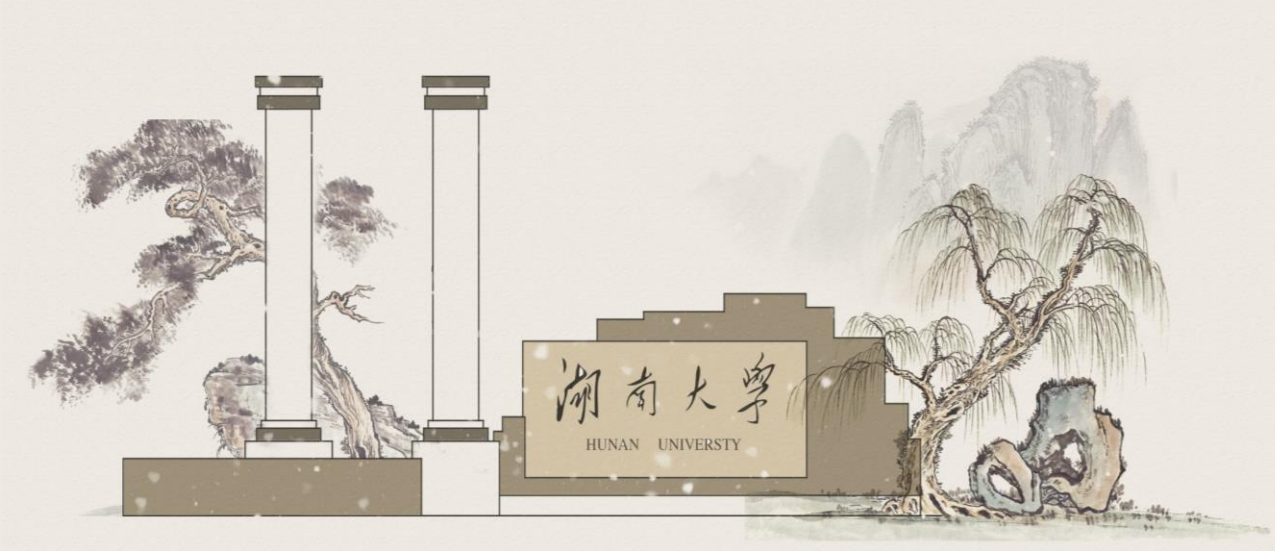
01

我们使用DNN来预测新CPU SKU上SPEC和Geekbench工作负载的性能。我们发现DNN比传统的LR更精确

02

在文献中，我们首次量化了这些广泛使用的套件的自相似性，通过使用DNN交叉预测套件之间的结果并通过比较它们的CPU排名。

02 本文方法



构建预测模型

预处理

- 对于离散数据，我们进行整数编码和热编码

- 动态工作负载特性：SPEC和Geekbench数据集中的每个条目都包含一个配置（SKU、当前频率、内存大小）、工作负载名称和性能（相对运行时，这是要预测的值）。然后我们通过SKU处理器编号从Intel处理器数据集中获取CPU特性

- SPEC2006存储库以秒为单位提供运行时。Geekbench 3提供吞吐量，例如GB/s，我们假设给定工作负载的总工作量是恒定的，并且我们使用吞吐量的倒数来估计运行时。我们首先用相同的配置来平均工作负载的性能。将所选参考机器的平均运行时间转换为相对运行时间。将相对性能标准化为平均值0和标准偏差1

模型选择

- DNN：与传统方法相比，DNN在数据量大、数据量大、扩展性好、不需要专门的特征工程知识的情况下具有优势。DNN的优点使其非常适合于性能预测场景。因此，在本文中，我们使用DNN作为预测模型。我们发现最佳超参数集总是相同的。因此，在所有实验中，我们都使用一个具有三个隐藏层和每层100个节点的DNN。学习率为0.001，超过300个训练时段。激活函数为tanh，优化器为RMSprop。因此，用于预测SPEC的权重数为5000万。。

- LR:LR在CPU性能预测表现良好，而DNN已知能够探索特征之间的非线性相互作用，并且在性能预测中比LR更准确。与DNNs不同，LR具有可解释的权重参数。在现实场景中，人们可以根据参数、精度和可解释权重的值来选择LR或DNN。为了使之成为一个公平的比较，我们以平均绝对误差（MAE）作为损失函数重新实现了LR。也就是说，对模型进行了优化，以获得最低的MAE。最小化MAE需要假设噪声分布为拉普拉斯分布。在我们的实验中，DNN和LR都具有L1正则化，权重为0.01。它们采用完全相同的训练集、测试集和特性。

结果度量准则

- 我们使用MAE作为精度的度量

- Top-K准确性：为了将MAE度量引入现实环境，我们使用预测模型来帮助选择SKU。我们预测性能并使用该预测对SKU进行排名。我们选择Top-K精度，它测量已知的最佳SKU是否在预测的顶级的TOP K 的SKU中

- 在没有预测模型的情况下，通常比较微体系结构的代码名（越新越好）、频率（越高越好）和缓存大小（越大越好。因此，我们在考虑这些因素的情况下构建基线。
  A 频率：根据频率对SKU进行排名。如果频率相同，使用缓存大小对其进行排序。
  B 缓存：根据缓存大小对SKU进行排名。如果缓存大小相同，使用频率对其进行排序。
  C 频率加缓存：根据频率和缓存大小加上权重对SKU进行排名。如果值相同，使用微体系结构代码名进行排序

03 性能预测

预测分析

- 新SKU预测

我们的模型可以通过训练现有SKU的SPC和Geekbech数据预测新的SKU在SPEC和Geekbech上的工作负载上的性能。我们采用了重复随机子抽样验证。我们从一个微架构中随机选择10%的SKU作为测试集。我们使用其余的数据作为我们的训练集。我们对每个微结构重复这个过程20次。

- 新工作负载预测

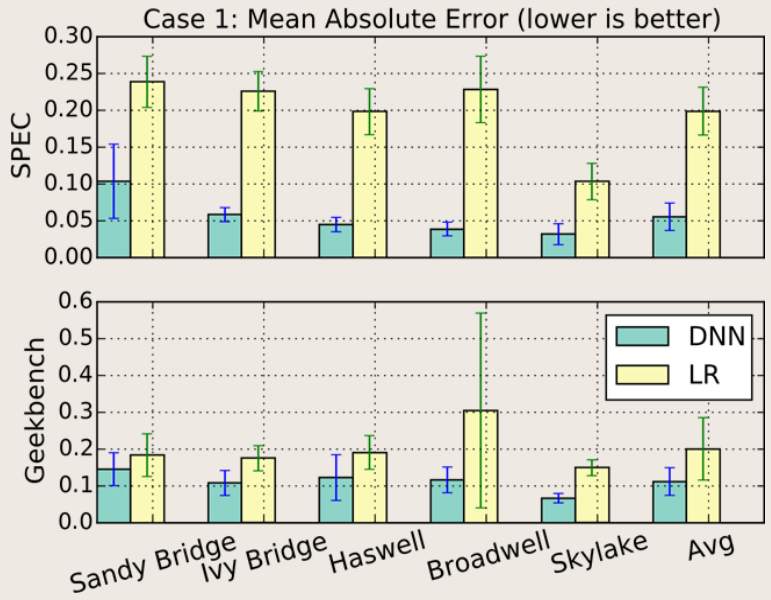
在测量了一个新工作负载在少数几个SKU上的性能之后，我们可以训练一个模型来预测工作负载在其他SKU上的性能。对于每个工作负载A，我们从整个数据集中删除工作负载A的数据。然后，我们将工作负载A的一组固定SKU作为测试集进行采样，并将工作负载A的其余数据作为训练池进行引用。从训练池中随机选取n个sku，并将它们的数据与剩余数据结合，形成一个训练集。重复这个过程20次。

- 套件之间的交叉预测

与新工作载荷预测方法大致相同，不同的地方在于使用SPEC的所有数据以及为所选工作负载使用几个SKU收集的数据来预测Geekbench中的一个工作负载。然后我们重复使用Geekbench和SPEC切换。

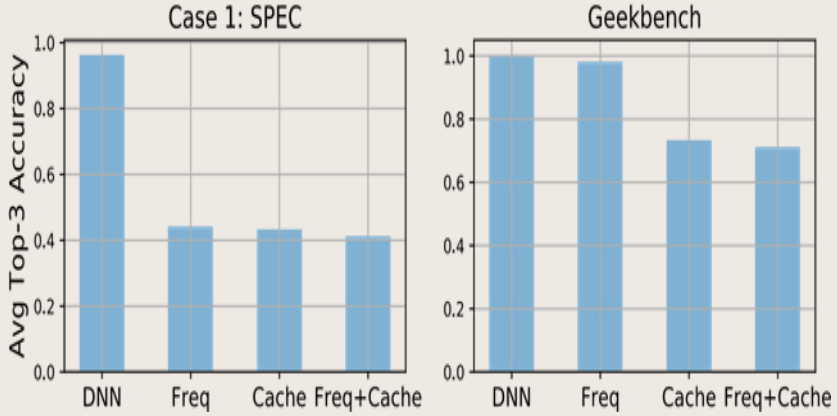
新SKU的预测

预测精度



预测新SKU上SPEC和Geekbench性能的结果。我们比较了线性回归（LR）和深度神经网络（DNN）模型。DNN的平均绝对误差（MAEs）总是低于LR。对于DNN，SPEC的平均MAE为5.5%，而Geekbench的平均MAE为11.2%。试验装置的标准偏差均在20%以上。

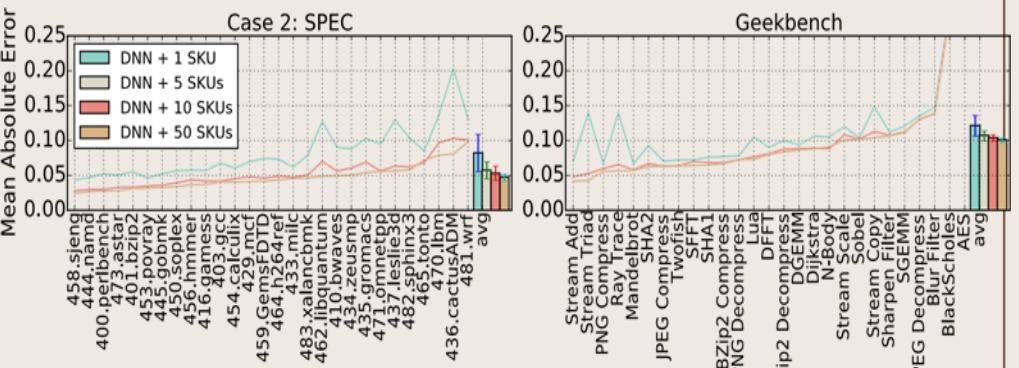
SKU排名比较



我们的DNN模型具有最高的前3个精度。我们的预测模型比简单地比较频率、缓存大小或两者的组合效果更好。按频率选择SKU比按缓存大小选择SKU更好，而且对于像Geekbench这样的微基准比SPEC这样复杂的工作负载更成功。

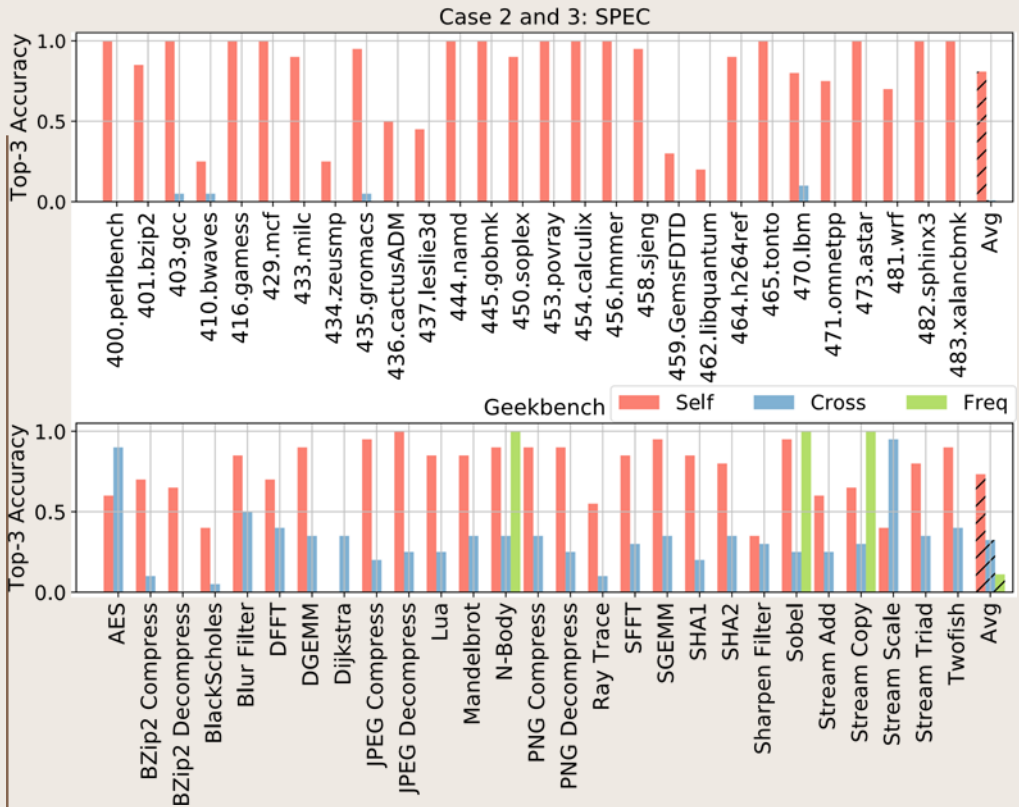
新工作负载预测

预测精度



通过在10个SKU上运行一个新的工作负载，我们能够预测它的性能。预测的准确性取决于工作量与训练集的相似性。即使不同，我们的模型也可以缩小置信区间。

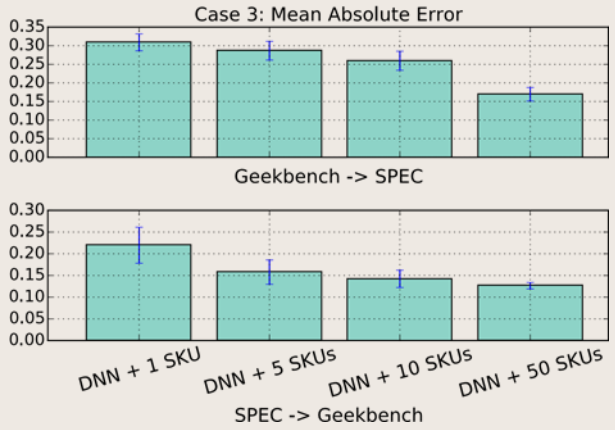
SKU排名比较



规范自我预测是寻找规范工作量最佳SKU的唯一合理途径。在大多数情况下，自我预测比交叉预测更容易。频率的精确度最低，因此在为新的工作负载选择SKU时，频率本身不应该是决定因素。

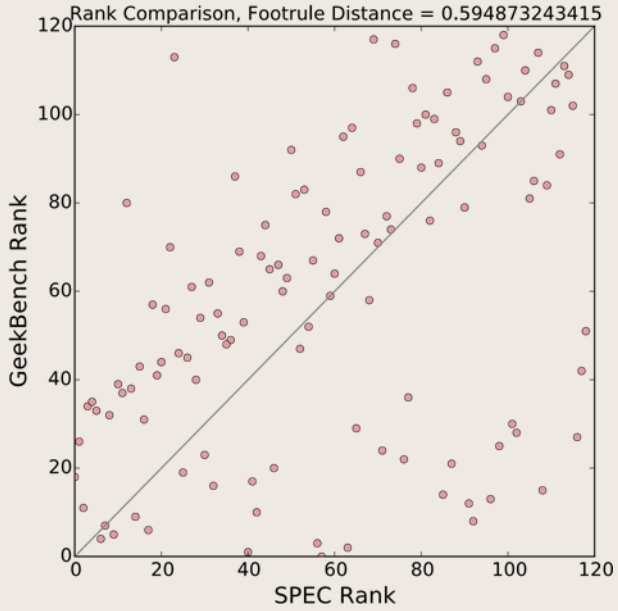
套件之间的交叉预测

预测精度



交叉预测比自我预测更困难。要预测新工作负载的性能，需要一个包含各种工作负载集合的培训集。基准套件往往是自相似的，因为每个都有特定的用途

基准套件比较

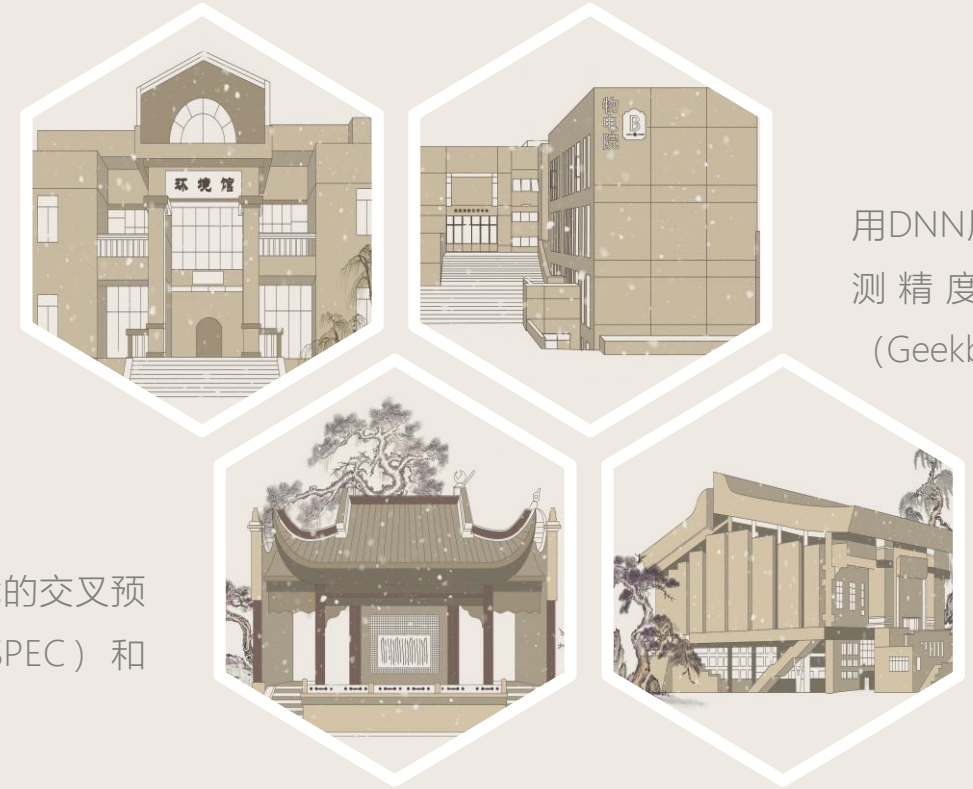


基于SPEC和Geekbench平均值的SKU排名比较。积分聚集在对角线上方，因为Geekbench将SKU的排名低于SPEC（给予它们更高的排名索引）。下面的水平集群显示，Geekbench很难区分具有完全不同规范行为的SKU。

04 总结

数据结果

用DNN展示了新SKU的性能预测。预测精度为5%（SPEC）和11%（Geekbench）MAE



用DNN展示了新工作负载预测。预测精度为5%（SPEC）和10%（Geekbench）MAE

用DNN展示了新工作负载的交叉预测。预测精度为17%（SPEC）和13%（Geekbench）MAE

同样精确度下，我们发现我们可以通过在10个SKU上运行一个新的工作负载来预测它的性能

结论

文交叉预测的精度低于套件内的预测精度，因为它们是如此的不同：预测SPEC的平均误差为25.9%，预测Geekbench的平均误差为14.2%。 基于平均Geekbench的排名与基于平均规范的排名不一致。Geekbench的排名并不意味着规范排名。对于基准套件中的实际工作负载，也很难得出任何结论。我们建议研究感兴趣的Geekbench或SPEC工作负载，而不是依赖于总分。

謝　謝　觀　賞

THANKS FOR WATCHING