# 从研究到实践

TL；博士：

* 做好准备
* 准备数据
* 在你的应用程序和最近的深度学习应用程序之间找到一个类比
* 创建一个简单的基线模型
* 创建可视化和调试工具
* 调整你的模型
* 端到端的训练、合奏和其他复杂性



莱斯利·史密斯是美国海军研究实验室的一名研究员。在深入学习研究方面，NRL并不是最重要的，但我真的很喜欢读史密斯的论文。我发现他们的贡献和Google/Facebook/Baidu上的人一样多。

本文介绍了使用深度学习解决新域或相邻域问题的推荐步骤。它没有引入新的体系结构，也没有提出令人费解的新训练方法，但我认为它的内容对于深度学习的成长同样重要，而且常常被研究人员忽略。

因为这篇论文的格式有点独特，所以我将把它作为一个注释的摘要列表来覆盖，其中穿插了一些注释，而不是使用我们的论文注释模板。

#### 第一阶段：做好准备

* 你需要深入学习吗？最先进的机器学习/统计技术在您的用例中是否表现得足够好？
* 成功是什么样子的？这是最好的定量陈述，理想情况下，与一个或两个您正在优化的指标。
* 你能走捷径吗？你能用更少的数据过日子吗？数据预处理，启发式？

很重要的一点是，通过给网络一个更干净的数据集（好的）与手动特征工程（坏的，这有点违背了使用神经网络的目的）来区分使网络工作更容易的方法



#### 阶段2：准备数据

* 你的数据来自哪里？你有多少钱？你有多大的信心？

我希望转移学习在这里得到更多的重视，而不是仅仅被提及。根据我在fast.ai课程中的经验，转移学习通常是最能平衡时间/精力和绩效的选择。

* 你的数据覆盖了整个问题空间吗？

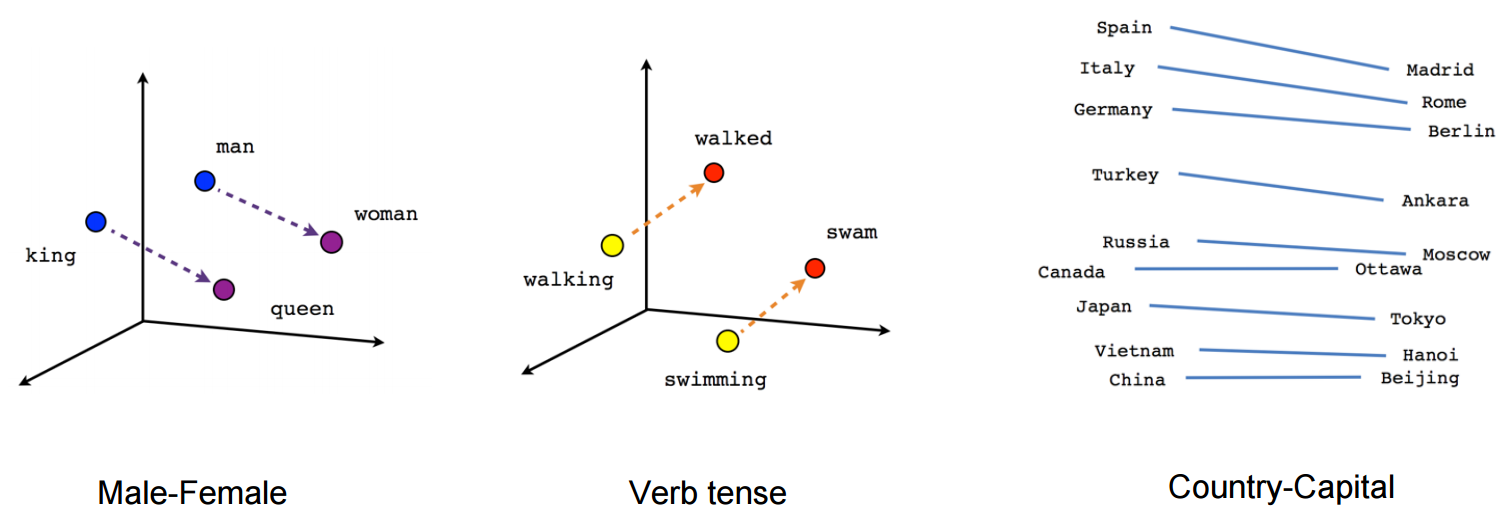
如果你试图预测数字0-9，请确保每个类至少有一个！可能看起来很明显，但当你进入大型分类器，它可以被忽略。

* 数据的类分布是否代表问题空间？

有时这并不理想，例如，如果你试图预测现实世界中0.05%的测试中出现的癌症，你会希望你的训练集有更高比例的阳性结果。只是要意识到这一点并做出相应的调整。

* Normalize和zero表示数据。几乎没有理由不
* 你能使用PCA或SVD技术降低数据的维数而不丢失太多信息吗？

对于新的深度学习应用程序，这可能是最重要的阶段。我认为，与神经网络体系结构等更令人兴奋的研究领域相比，数据的重要性仍然没有得到足够的重视。希望钱能起作用，这会改变。



#### 阶段3：在你的应用程序和最近的深度学习应用程序之间找到一个类比

* 深度学习的大多数低调应用已经被研究者解决了，而且你的灵感可能从已经存在的东西中得到灵感。作者建议按类别分类：
* 图像分类：CNNs
* 时序/时态数据：RNN（LSTM/GRU）
* 决策：深度强化学习
* 使用此步骤中的代码或摘要来通知下一步的方法

再一次，作者几乎是一路在这里转移学习；我想知道为什么他们不采取更多的步骤，并建议不仅找到代码，但找到重量和偏见，你可以开始自己的模型。

#### 阶段4：创建一个简单的基线模型

* 使用公共损耗函数和安全超参数，使网络尽可能小，具有最少的层和参数。

在他的fast.ai课程（强烈推荐btw，Paper Club同时经历了这两个部分）中，Jeremy Howard通过构建一个只有输入层、一个隐藏层和一个输出层的网络来解决问题。一开始这听起来可能没什么用，但一次又一次，这已经足够让人看到趋同了。我个人认为这种方法非常有益。

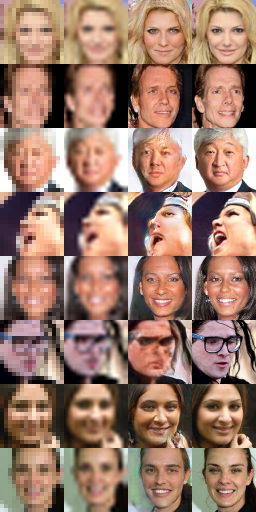
* 对训练数据的子集进行训练，以缩短反馈周期。

只要有足够的数据让模型收敛，这一步就足够了。

* 单元测试您的网络

我从来没有见过一个单元测试过的神经网络，这让我想知道为什么A和B不存在什么工具。

* 利用TensorFlow或Pythorch这样的框架



#### 第5阶段：创建可视化和调试工具

* “编码一次，测量两次”。

从第1阶段到第2阶段，你应该对你期望得到的结果的类型有一个非常明确的概念。现在是时候仔细比较这些和你的实际情况了。检查你的架构、重量、偏差等是否符合你的期望。我个人发现自己在某个时刻打印出了网络中几乎所有重要张量的核形状。

* 高偏差与高方差：高偏差意味着你的网络正在收敛到错误的极小值或局部极小值，可以用更大的网络来解决。高方差意味着你的人际网络根本没有收敛，这表明你的训练数据不足。
* 错误分析与烧蚀分析：错误分析是您的典型损失，代表您的网络性能和完美性能之间的差异。烧蚀分析是建立一个基线的过程，在你的网络中一次改变一件事情，并注意改变后的网络与基线的比较。两者都有用。
* TensorBoard与TensorFlow捆绑在一起，是推荐使用的工具

#### 第六阶段：调整你的模型

* 改变一切！你应该修补你的网络的每一个方面，并记录它如何影响性能：架构设计、深度、宽度、路径、权重初始化、学习率等。

在这个过程中，当你的训练精度不断提高，但你的验证精度却停滞不前或下降时，你可能会发现自己在某个点上面临着过度拟合的问题。这是一个典型的机器学习问题，当你看到它时，首先要做的是拍拍自己的背，因为你有一个正在学习的网络！然后，就按照既定的行动手册来缓解它：

* 添加更多数据
* 使用数据扩充
* 使用通用性好的架构
* 添加正则化
* 减少体系结构的复杂性。
* 损失函数是网络中最重要的组成部分，因此请务必深入思考什么类型的损失函数是合适的，并测试一些可能符合您的标准的损失函数。
* 这个阶段是完全开放的，如果没有某种程度的自我控制，最终可能就是这样。一定要经常回顾你在第一阶段的既定目标，当你达到目标时强迫自己停下来，让你脑中的声音安静下来，唠叨着你还没有尝试过的新的损失功能。

#### 阶段7：端到端训练、合奏和其他复杂

* 还有更多！作者指出，如果你对自己的网络表现感到满意，那么完全没有必要使用这些工具，但建议使用两个最有价值的工具：端到端的培训和整合。端到端的训练意味着试图将一个包含多个部分的系统整合成一个神经网络，该网络接受原始输入并产生所需的最终输出，即使它们的格式不同（如语音到文本）。集合是训练具有不同随机权值初始化的多个模型，并对其结果进行平均或加权的过程，以减轻单个模型的任何偏差或奇异性。

And that’s that! Just follow these 7 simple, straightforward, easy, clearly-defined, no-confusion-at-all, steps and you’ll be on top of the world 😉