

订阅DeepL Pro以编辑此演示文稿。  
访问[www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)，了解更多信息。

好吧，我得知你们今天已经张贴了演讲稿，是否还有另一份材料也应于今天提交，还是只有海报？天哪好吧，我会接受的。我不会太在意。不过，我想我会花两三分钟时间和你们闲聊几句，以防还有人想加入我们。你要记住不要开着麦克风跑 因为麦克风显然是固定在房间里的因此，如果你走得太远，它们就会发出某种警报，然后技术人员就会追着你跑，以防你想偷麦克风或只是不小心把它带到了你的办公室。所以我必须尽量避免走得太远。我不知道我能走多远。好了，考虑到这是最后一堂课，我们会轻松一点。但我们今天还有一个模型要讲。我想，大多数人其实都在关注这个模型，我们也期待听到更多关于它的内容，因为它是新闻中经常出现的模型之一，尤其是当我们谈到深度学习这个概念时。因此，我们今天将讨论神经网络，我将简要谈谈深度学习，它在这种情况下究竟意味着什么，在哪种情况下有用，或者什么时候有点用。是的，也许被夸大了，或者没有想象中那么有用。所以，这又回到了我们在第一讲中提到的观点，即每个模型的好坏取决于你使用它的环境。因此，你遇到的每个问题都取决于你有什么样的数据、什么样的目标、什么样的关系，你认为数据中可能存在什么，你正在做什么，试图预测什么，取决于你对所有这些问题给出什么样的答案，不同的模型会给你更好的拟合效果、不同的拟合效果、计算效率更高，或者根本没有任何意义。举例来说，我看到深度学习被应用于数据集很小、线性关系很简单的问题。我们会看到为什么这没有任何意义。因为神经网络的最大优势，也是我希望你在听说和考虑应用它们时要牢记的一点，就是它们本质上是非线性的。这是神经网络一个非常非常大的优势。我们会知道为什么，它们的结构到底是怎样的，但从这个意义上讲，它们与支持向量机以及 K 和 N 随机森林决策树类似。所有这些方法都是非线性的。因此，线性关系其实并不罕见，但纯粹的线性关系并不常见。如果你确实遇到了一种非常直接的线性关系，在大多数情况下，线性回归就是你所需要的，因为你也会看到，许多更复杂的方法，比如神经网络，只是一种类固醇的线性回归。因此，我们会发现神经网络采用了线性回归的概念，并开始以各种奇怪的方式将其结合起来，通过非线性函数来推动它。这样就得到了非线性结果。但它们使用的很多概念你都很熟悉。因此，如果你了解线性回归，那么神经网络就是在此基础上更进一步，是一种让它们变得非常非线性的方法。上周我们讨论了支持向量机。希望你还记得。我为自己解释核技巧的方式感到非常自豪。我觉得我的想法很有道理。也许我应该成为 YouTube 或 TikTok 的教授之类的。是的，我不知道这是不是个好主意，但你可能会在几年后看到我这么做。所以，你还记得支持向量机吧，这种寻找线性决策边界的想法，可以是一条线，也可以是一个超平面。通过使用内核技巧，还能划分数据中的非线性关系。这种通过线性空间推动某种函数的整体思路，将数据的某些部分推入一个新的维度，即允许你使用线性决策边界的额外维度。在这种情况下，我们在讨论最大边际分类器，更一般的优化分类器，我们在讨论支持向量分类器和支持向量机。你一定记得我说过，支持向量分类器基本上就是 SVM，但通常我们在有线性关系时才会使用这个术语。因此，严格意义上的术语应该是线性支持向量分类器。但没人真的这么说。你只会说它是一个 SVM，使用线性内核或根本不使用内核。本周我们将讨论神经网络。我们将讨论单层前馈网络，这是最简单、最基本的结构。我们将讨论深度多层网络，以及它们与深度学习的关系。我们还将讨论网络正则化和调整。因此，请记住，这是一个非常基础的神经网络入门讲座。在此基础上，还有更先进的方法。例如，你可能听说过循环神经网络。工程或长期短期记忆网络，所有这些更高级的结构。但为了使用和理解它们，你首先要了解基本结构，我们会把重点放在何时使用它们，以及如何在不同情况下使用它们。看，这很值得。先聊几分钟，再等一会儿。时机不错。我之前说过，神经网络本质上是非线性的，因此既可以用于分类，也可以用于回归。我们会看到具体的方法。但你可以把它想象成回归意义上的线性回归，然后把它想象成分类意义上的逻辑回归，因为我们会使用一个非常相似的技巧，基本上是一个 sigmoid 函数或与 sigmoid 函数非常相似的东西来获得不同类别标签的概率。这样，我们就可以将它们用于分类。所以，你可以把它想象成线性回归。逻辑回归就是大回归。神经网络这个名字实际上与大脑有关。因此，我们正试图模仿大脑的运作方式。我们使用的很多术语都与神经学类似。因此，我们将讨论大脑神经网络中连接的节点。话虽如此，但我不是神经学家。我不知道大脑是如何工作的。我想大多数生物学家都不知道。所以，不要以为你的大脑就是这样工作的。它可能不是。这只是一个极其简化的版本，在理论上，对于非常简单的任务，它可能看起来非常相似。令许多人惊讶的是，神经网络真正出名是在上世纪 80 年代。那是 50 年前的事了，当时人们对神经网络感到非常兴奋。所以每个人都认为，哦，这是下一件大事。它就像大脑一样。所以，这肯定就是人工智能，就像峰值一样，我们都应该这么做。然后很快，它们就失宠了。所以我认为，在 90 年代，2000 年初，没有人在使用神经网络。原因很简单，神经网络非常昂贵。它们的计算成本非常高。稍后你就会明白，为什么当我们有了 SVM 和随机森林之类的东西时，它们的效果一样好，有时甚至更好，而且更便宜。所以，如果神经网络不能很好地工作，即使它看起来像大脑，我们为什么还要用它呢？因此，我们很长一段时间都没有真正使用它们。我觉得有两个原因。其一是，它们需要大量的修补工作。这与计算成本有关。神经网络有很多参数。我的意思是，即使是小型网络，也有成千上万个参数，而 SVM，正如你之前看到的，它有两个参数，一个是你想要选择的内核，另一个是你的评估或成本计算参数 c。因此，与神经网络相比，调整它们来适应你的问题要容易得多，因为神经网络需要调整很多小参数，而且成本更高。在很多情况下，SVM 和随机森林的表现都要好得多。所以最近，我想最近我得把这个词改成明年，因为现在已经不是最近了。人们又开始推崇这种方法。主要原因是我们有了更多的数据，我们有了更好的计算，我们有了更好的计算架构。训练神经网络变得越来越可行。我认为另一个原因是很多开源编码社区开始推动这方面的编码工作。因此，在我们的实验室里，我们将主要使用基于 TensorFlow 的 Keras 库或环境，并将其开源和公开，作为供人们使用的编码环境和库，这使得神经网络突然变得非常容易为人们所使用。因此，你会发现，只要是结构相对简单的神经网络，比如简单的前馈网络，就很容易实现。基本上，你只需指定不同的层级、多少个节点，然后只需五行代码。然后你就拥有了一个完美的神经网络来解决你的问题。因此，这种环境、这种库确实促使人们开始修补它们，并再次尝试使用它们。是的，它们在 2010 年左右重新出现，也许比 2010 年稍晚一些，因为人们对深度学习的兴趣增加了。新架构我们有一些成功案例，尤其是在图像分类和视频分类领域。我认为这是神经网络真正大显身手的地方，这些问题极其复杂，而且在语音和文本建模方面也是如此。因此，人工智能界，尤其是自然语言处理领域，往往对神经网络等非线性复杂方法感兴趣。让我们来看一个。我之前说过神经网络。因为它的核心它与线性回归非常相似。你会明白为什么。假设我们有一个输入向量这就是你的数据我们正在寻找某种非线性函数，试图用它来预测某个输出值。所以这是一个回归问题。我们有数据我们把它放进去我们想要一个数值现在神经网络的结构是这样的你可以看到黄色部分是输入层这是你的数据。然后是输出层。也就是最后的 Y 值和某种函数。新网络的有趣之处在于中间的这一层，以及它是如何连接到这两个站点的。因此，你可以看到这其实相当复杂。我们的每个输入值都要经过每个隐藏层、隐藏层中的节点，然后通过某种非线性函数转化为我们的输出。你还可以看到，这些箭头是有方向性的。所以，我们说的是前馈网络，因为我们通过网络从左向右前馈数据。在优化步骤中，会有一些回溯和反向传播。但在这个非常简单的想法中，我们获取数据，将其推送到网络中，最后得到一个值。下面是一个例子。我们有四个预测因子。这就是我们的四个值。我们输入的值。我们把它们输入进去我们有 K 个隐藏单元，这里有 5 个。这是你可以选择的数量所以你可以选择你想要的层中有多少个隐藏单元。单元越多，就越复杂，对于更复杂的非线性关系就更好。单位越少，成本越低，越简单。更适合简单的关系。这与我们过去看到的其他模型非常相似。因此，如果我们实际看一下计算过程，就会更清楚为什么我们说或为什么我说它的核心与线性回归相似。所以你可以看到，我们仍然有一种线性组合的因子。因此，你可以再次看到这些向量，它们会让你想起线性回归的类似想法。我们正在估算某种系数。这里有我们的输入值 x。我们试图用它来估算某种输出值 y。而问题在于，我们要在这些线性组合中分层。例如，我们有隐藏笔记 a，这是我们的五个隐藏节点。在每个节点中，我们都要进行线性运算。因此，每个向量或每个值都会被输入到这些节点的神经元中。然后通过线性组合进行计算，再通过这些计算进行前馈。因此，这个想法是相对线性的。不过，我们在每个节点中都使用了非线性激活函数。这就是非线性的作用所在。因此，这种非线性激活函数可以是如sigmoid函数或多项式函数。因此，核心在于它必须是某种应用于输入特征线性组合的非线性函数。这就是核心。它与非常复杂的线性回归嵌套系统的区别是什么？这就是核心所在。最后，我们把这个隐藏层，也就是最后一个隐藏层输入到输出层，然后预测出某个值。所以，请记住我们这里的 A，也就是我们所有的隐藏层，所有的隐藏节点。因此，每个小 A 都有一些参数，因为我们将所有这些 A 组合在一起，预测一个值。所以，是的，它类似于线性回归模型。但我们有五个神经元。而这些神经元又是输入数据线性组合的非线性变换。所以，这就是它变得如此复杂的原因。因为突然之间，我们已经有了很多参数。你可以在这里看到更好的参数。最后是我们的输出层。我们有所有这些 W，你可以把它们看作权重。你可以把它们看作是每个节点中 X 值的权重或比例。所以它并不总是相等的。所以这一点也很不明确。你也不知道到底发生了什么。但你知道每个 A 对我们的每个 X 都有一定的比例和影响。因此，你可以看到，对于一个非常小的网络来说，已经有很多参数需要估算了。记得我们说过，我们有四个输入值、五个节点和一个要预测的值。所以我们总是在为一个非常小的网络估算大量参数。我之前提到过，这些激活函数，也就是这些线性组合中的非线性部分，可以是，例如，一个 sigmoid 函数。你还记得 sigmoid 函数吧。我们就是用它来做逻辑回归的。我们就是用它来把这些线性函数转换成概率的。如今，我们通常使用 ReLU，它代表整流线性单元激活函数。它们与 sigmoid 非常相似，所以不用太在意两者的区别。我在这里放了一个区别来解释。你可以看到它的原理基本相同。在达到某个临界值之前，数值可以为零，然后迅速上升到临界值之后的数值。因此，它类似于绿色的 sigmoid 或 sigmoid，只是更平滑一些。主要区别以及我们现在使用射线的主要原因是，它的计算存储成本更低。因为你可以看到这里所有的值都是零。然后在这里，我们将继续把所有这些值都存为 1。因此，这里的这些值的存储量更少。因此计算成本更低。而且它的总和效果几乎一样。这就是为什么我们要使用这个函数而不是 sigmoid 函数的原因。话虽如此，如果你使用 sigmoid 函数，其实也没有太大关系。有些人仍然保留着那些老方法。正如我之前提到的，这是一个发展相对迅速的领域。所以，当我说起 2010 年的时候，我觉得那已经是很久以前的事了，可能感觉就像昨天一样，但那已经是很久以前的事了，而且我们还在进一步发展。所以，也许五年后我们又会改变主意。但现在，我们坚持这样做。那就是他了是的总之，在这种情况下，模型是一个有五个隐藏神经元的网络。这里有这样一个术语。你也可以说节点、单元、神经元或其他任何你喜欢的东西。我们通过计算 x 的五个线性组合，并通过某种激活函数将其压扁，从而得出五个新特征。然后，最终模型与这些新的衍生变量呈线性关系。请记住，这又是线性的。所以，你基本上是线性组合的非线性变换的线性组合，如果这能说得通的话。这就是非线性的由来。实际上，我这里有一个计算实例。我想，之所以要举这个例子，主要是因为我想从数学的角度向你展示为什么它实际上是非线性的。这里有一个非常简单的例子，我们只有两个输入变量。X1 和 X2。我们有两个隐藏单元。这就是中间的神经元。我们称之为 H1X 和 H2X。我们定义某种非线性转换压扁函数或激活函数。我喜欢称它们为压扁函数，因为我觉得压扁线性组合是个有趣的想法。比方说，我们已经估算出了所有不同的参数。所以你会发现，估计这些参数的计算成本相当高。假设我们对所有参数都进行了估算，因为我们想看看这些计算结果是什么样的。我们知道这是我们的非线性函数 g。记住，这基本上是每个变量对神经元的影响。我们正在计算 H1 和 H2 的权重。这就是我们的两个隐藏神经元。所以，你会发现，如果我们真的这样做了，做完所有这些漂亮的计算步骤，我们只需插入我们的贝塔值，我们已经说过，我们已经神奇地估算出了贝塔值。最后你会发现，实际上得到的是 x1 乘以 x2，这当然是非线性的。所以你可以看到，这两个非线性变换的总和给了我们这两个变量之间的交互作用，而不仅仅是它们的简单线性组合。因此，如果我们要估算 y 值，我们就可以通过这两个变量之间的交互作用来估算，而不仅仅是线性回归中的 x 一加 x 二。因此，通过这些非线性激活函数来转换线性组合，就能捕捉到这些非线性因素和非常有趣的交互效应。是的，我们刚刚说过，这些参数会神奇地出现。但事实当然并非如此。我们实际上必须对它们进行估算。好消息是，这个过程和你已经习惯的一样吗？因此，我们仍然需要通过某种误差损失函数来进行估计。在回归问题中，我们仍然要最小化误差。因此，这仍然是我们的真实值减去估计值。我们将这些误差平方。我们将它们相加。这与我们在分类问题上所做的是一样的。也是一样的。所以，你仍然可以只估算出分类的误差。让我们来看看这看起来像什么。多层次，因为我认为这是有趣的部分。你现在看到的只是一个单层神经网络。你已经看到这是一个有点复杂的东西，但现在大多数神经网络都有不止一个这样的层。也就是说，有不止一个隐藏层。在很多情况下，你有很多单元。所以每个层中也有很多节点。因此，这是一种概括。多层网络可称为深度神经网络。深层指的是有多少隐藏层的深度，因此，人们对于需要多少层才能称之为深层网络会有不同的看法。很显然，因为没有人能在这一点上达成一致。然后，也有人会转过头来说，是的，这是一个非常深的神经网络，还有一个更深的神经网络，等等等等。但一般来说，层数指的是深度。一般来说，只要层数超过一层，就可以称之为深度神经网络，这取决于你对深度的定义。因此，从理论上讲，我们刚才看到的是单层。你已经可以对大部分复杂性和大部分非线性进行建模，但在某些情况下，使用节点较少的多层比使用节点较多的一层更容易。因此，在很多情况下，为了真正达到你想要的复杂程度，你必须有一个非常宽泛的层，而不是一个很深的层。因此，与其拥有大量的节点，你可以拥有更小的层，然后只需堆叠其中的几层，并通过所有这些层推送你的数据，这样就会产生类似的效果。在某些情况下，它还能为你的解决方案带来更好的结果。因此，我确实有这样一个多层网络的数据。我用一个非常著名的神经网络例子来解释这个问题，那就是图像分类。我之前提到过，神经网络有一个非常专业的细分领域，它们在该领域的表现非常出色。其中之一就是时间序列。因此，时间序列预测实际上是神经网络表现非常出色的领域之一。你经常会在商业环境中看到它们的身影，例如，在金融领域，你有很长的时间序列，而你正试图预测时间序列的下一步，例如，股票市场预测，新网络在这类任务中表现相当出色。另一个领域是图像分类。这是一个非常有名的例子，我们试图预测笔迹。特别是，我们试图通过观察某人写在纸上的数字的图像来预测这些数字。在这里，我给你一个相当大的两层网络。我们有两个隐藏层。一个有 256 个单元，另一个有 128 个单元。现在你问我为什么是这些数字，也许你能想象出我的答案是什么。在这种情况下，你必须进行交叉、验证，并寄希望于最好的结果。因此，与我们之前的五个数字相比，这是一个相当大的数字，对吧？所以，我们会看到它的计算成本有多高。我还说过我们有数字，所以我们会有十个输出变量。这实际上是一个分类问题。我们并不是要预测一个聚合值。我们要对 0 到 9 的数字进行分类，看看这些笔迹指的是什么。因此，这种多任务学习系统可以通过神经网络一次性完成。我们正在讨论 SVM，以及如果你有两个以上的类别，它们是如何变得有点棘手的，所以你可以使用它们，但它通常适用于二进制情况逻辑回归，同样的事情。逻辑回归。一般来说，二元输出可以堆叠组合，用于多类预测，但目标只有一个。神经网络的多分类预测效果非常好。这就是它的样子。显然，这不是一个完整的网络，因为我无法为你画出那么多注释，但这实际上就是它的结构。然后这将有 200 个。这些蓝色神经元有 56 和 128 个。所以，你可以看到我们有输入层。这是我们先训练再测试的数据。就像你平时做的一样。然后我们有一个输出层，我想要九个数字。所以，这是 0 的概率和这是 9 的概率，以及介于两者之间的所有概率。因此，我们试图预测一个概率，正如我之前所说的线性回归，指的是一种类似于分类神经网络的回归网络逻辑回归。你还可以看到我们的权重。记住 W 是输入对每个节点的贡献。然后是我们的赌注。这就是我们如何将其转化为输出值的过程。除此以外，这与我们刚才所做的非常相似。最大的区别在于，你可以看到我们首先开始将输入数据输入到我们的每一层，也就是第一层的每一个节点，就像之前一样。但现在，这些数据的输出不是作为数字的直接输出，而是作为下一层的输入。这就是关键的区别。在这里，我们开始将非线性因素相互叠加。因此，这是一个嵌套的非线性系统。最后，我们再做同样的事情。实际上，我们的输出层就是我们的非线性变换层。在这种情况下，我们会使用 sigmoid 或类似的东西，因为我们需要概率。因此，第一层看起来很熟悉，因为这和我们之前做的是一样的。这次我们有了权重。我用一个小 1 将其索引，表示这是我们要输入的第一个隐藏层。我们有正常的线性组合和某种权重系统。你可以把它们想象成线性回归中的贝塔、输入数据 X 和非线性变换函数 g。记住 k 是这一层的节点数。第二层将第一层的输出作为输入。因此，在这种情况下，你可以看到这基本上增加了一点复杂性，但看起来并不太糟糕。目前唯一不同的是，这个 a 就是这个词。所以我在这里写的是简化版，但你可以想象，这里的这个 a 其实就是整个表达式。这就是为什么我们说这是一个嵌套的非线性系统。非线性变换 G1A 和非线性变换 G2 基本上是这样的。否则就是同一个系统。我们将 WS 索引为 2，因为它们现在是第二个隐藏层 a two 的权重。所以，重要的是，第二层中的每个单元仍然是输入向量 x 的函数。所以，你仍然将 x 输入第二层，但你输入的是 x 线性组合的非线性变换。所以，它是 A 的函数，但这些都是 x 的函数。所以，如果你仔细想想，我刚才说的是，如果你把这整个项放到这里，即使在第二隐藏层中，你也会把 X 作为输入数据。如果你有越来越多的层，情况也会是这样，所以不必拘泥于两层。理论上，你可以有数百个层，而且它们都是嵌套组合。这样就可以不断地叠加下去。你也可以看到，这将导致 X 的复杂变换。只是现在的变换更加复杂了。这也是神经网络是黑盒模型的原因。老实说，一旦你开始将数据输入这么多层非线性变换、不同的权重、每个节点中不同的 x 数量，你就完全不知道发生了什么。在某些时候，你根本无法追踪 X 是如何通过所有这些步骤进行转换，并最终得到 Y 值的。因此，神经网络是一种黑盒模型，你并不确定它们为什么能工作，但你知道它们工作得有多好，因为你仍然可以在最后计算出模型的误差。唯一能稍作转换的方法就是查看特征的重要性。因此，我们在讨论其他模型时也提到了这一点。但对于神经网络来说，这才是真正的关键概念，或者说是在一定程度上理解发生了什么的关键方法。在我看来，特征的重要性可以稍微考虑一下。就像我们在讨论 PCA 时，我们讨论的是载荷。那么，在我们构建的主成分中，每个变量的载荷是多少？我喜欢用类似的方式来思考特征的重要性。模型中变量的载荷是多少？那么，这些变量中的每个变量对最终构建的 Y 有什么样的影响？这一点你无法确定，但可以给你一个大致的概念。因此，我们将在实践过程中进一步研究特征的重要性。计算机实验室这就是明天计算机实验室的主题。天哪，我今天太兴奋了。睡眠不足好了，让我们看看我们的数字示例会是什么样子。我们说我们有一个 258 或更高的两层网络。285我不记得了，还有128之类的。这就是我们的两个隐藏层。我们试图对数字图像进行分类。有人写了数字而我们的神经网络正试图预测 或者说试图理解和模拟这些数字是什么还有。我们的方法就是查看这些图像在这种情况下，他们测量的是 28 x 28 像素。因此，这是大小为 28 x 28 的图像，这样每个样本就有 784 个像素。所以，你已经可以看到，这是大量的数据。这也是为什么神经网络在图像分类中效果如此之好的原因。因为这些数据量很大。假设我们的第一隐藏层有 284 个单元。然后我们计算权重矩阵的权重 w1，这样就有 285 个像素加上 1 乘以 200 英里，也就是 256。可能是因为我不会数数。我们之前说过什么？一个图层的大小是多少？256 MB所以应该是 256。那么 285 乘以 256。这一层的单位数就是 200,960 WS。这是一个很大的数字。请注意，我们这里的数据是 785，而不是 784，因为我们必须考虑截距。请记住，在线性回归中，当我们将整个数据输入其中时，有时模型会告诉你，我不能这样做，因为我需要截距来实现实际功能。同样的道理。你必须在其中加入截距。因此，这就像你的随机项或常数项，基本上就是你的模型如何锚定自身。因此，我们将这些元素中的每个节点都设置为 184 个，并馈入第二个隐藏层。因此，我们有 128 个节点。这样，我们就有了第二个双层，大小为 32 896 个元素。这就是第二层。现在我们还有输出层。这是实际的预测部分。基本上，这与逻辑回归非常相似，我们有十个而不是一个响应。我们的方法是，针对每种可能的反应，计算出十种不同的线性模型。这样我们就能找出所有更好的值。因此，我们有 1290 个贝塔值，用于最后的估算步骤。所以。我们之前说过，我们试图预测概率。这与逻辑回归非常相似。我们需要某种压扁函数，将其转化为概率。在这里，我们使用了一个软最大激活函数，它看起来与 sigmoid 非常相似。所有这些只是形状上的细微差别。我们之所以使用 softmax 而不是我不知道的任何其他函数，是因为人们刚刚开始使用它，然后其他人也采用了它。现在大家都在用。好吧，所以别质疑太多。有时候，研究的工作方式很神秘。它是有效的你可以用逻辑回归的方法来思考它好的，我们正在寻找类概率。样本有多大可能属于1号或6号之类的类别我们使用这个激活函数计算这些概率。在这种情况下，我们的目标是建立一个分类器。所以我们试图预测每个类别的估计概率。然后我们将图像分配到最有可能的类别中。因此，概率最高的人就能获得样本。他们基本上是在投票决定谁最有可能获得样本。另外，我还想说的是，我们如何实际估算这些系数？这些系数。这就是。在这种情况下，因为这是一个分类问题。我们会使用一种叫做交叉熵的方法。如果你以前没听说过交叉熵，你可能听说过最大似然法。那么最大似然法。交叉熵基本上是同一枚硬币的两面。所以它们基本上是一个非常相似的概念。交叉熵背后的基本概念是，我们正在研究两个概率函数之间的差异。所以在这里，我们基本上是在研究所有类别概率函数之间的差异。天哪我们试图将其最小化。所以我们要找到最有可能的类别。这个概念与线性回归拟合时误差最小化的思路非常相似。因此，这都是为了最小化某种成本函数，以便找到参数最匹配的模型。参数是一个小螺丝钉，你可以通过改变参数来改变模型的拟合程度，显然，你希望拟合的模型能与你的示例数据非常接近，但又不能太接近，因为在这种情况下，你会得到一个过拟合模型。那么，我为什么要告诉你这些数字呢？因为我想让你看看这个数字。200,000 235,000 和 146 个参数，用于估计你的图像分类问题。所以，我想我不必告诉你这是一个很大的参数，而且计算成本很高。我想让你思考的另一件事是，我想我们已经简单提到过了，但可能还不够详细。为了估算如此多的参数，你需要大量的数据。通常情况下，你希望用于训练的样本数量高于预测器或估计器的数量。你有参数，不管你要估计什么。因此，你通常需要更多的数据。然后你必须进行估计。这并不总是可能的。有些模型对此比较稳健。例如，我认为 k 和 n 模型对较小的数据集比对较多的预测因子更稳健。但通常情况下，你需要大量数据。如果你有很多参数，这就是你应该寻找的基准。因此，为了估算出超过 20 万个参数，你也需要大量的数据。为了训练大量的数据，你需要大量的计算能力。因此，所有这一切都共同导致了神经网络的运行成本如此之高。好了，我想这是个休息一下的好地方。我们先休息十分钟，然后再回来看第二部分，在第二部分中，我们将进一步讨论神经网络的拟合问题。最后，我还想用十分钟来复习一下考试。鉴于这是最后一堂课，你们还有什么问题吗？好的，我们十分钟后在这里碰头。我要走了我先走了我们要重新开始吗？接下来的半个小时左右，我们会花更多的时间来研究我们的神经网络，然后会有很多时间来回答关于考试的问题，以及其他任何你想问我的问题。我们知道什么？好的，我们刚才已经说过了。如果我们想象一下，一开始我们有一个非常简单的单层神经网络。你还记得，我们有一些更好的参数，这些参数被输入到最后的输出层，我们还有一些 w 参数，这些参数被输入到我们的输入层和隐藏层之间。因此，你可以看到，我们将再次面临某种误差最小化问题。没错。所以我之前说过，在分类问题中，我们试图最小化交叉熵。而如果是回归问题，我们又会用最小二乘法作为优化方法。因此，它与线性回归非常相似，但却是非线性的。因此，我们正在研究一种非线性最小二乘法。思路还是一样的。我们仍然试图将误差最小化。我们的误差仍然取决于我们的估计值有多大或多远。这就是我们对 x 的估计值与已知真实值 y 的距离。我们仍在对所有这些误差进行平方运算。我们还是要把所有误差相加。最大的不同是，你可以看到我们突然有了很多参数，我们可以稍微调整一下，找到最佳拟合。在过去，这只是线性回归的较佳参数。你只需改变更好值。现在，你还可以改变 W 值和所有参数。另一个很大的不同是，正如我们之前看到的，x 的 f 是非线性的。因此，这就是我们一开始使用的公式，我们使用非线性压扁函数 g 来处理权重和数据输入的线性组合。所以想法非常相似。一个很大的问题是，这个目标函数看起来很简单。正如我刚才所说，它与线性回归相同，对吗？但问题是，这些参数都是嵌套的。因此，实际添加的层数越多，参数的嵌套组合就越多，这基本上导致问题变得相当复杂。其中最大的问题是，我们正在研究的参数最小组合函数是非凸函数。这意味着什么？实际上，我们很难找到一个完美的最小值，一个完美的参数组合来最小化我们的 ABA 函数。因此，如果你想象一下这里的参数组合。我们所有的参数。因为我们都是在同一个目标下进行估算的。没错所以，如果我们有所有这些参数，并且我们正在寻找最小值，也就是我们损失函数的最小值，你可以看到，基本上有两个可能的最小值，你可能会达到，而且一个比另一个更好。因此，我们认为存在多个局部最小值，很难找到全局最小值。因此，我们需要找到所有这些参数的最佳组合。这是我们面临的一大挑战。我们可以用两种方法来避免陷入局部最小值，永远找不到完美的参数。一种是我们所说的慢速学习，并与早期停止相结合。因此，我们使用一种非常缓慢、渐进的方式来寻找最佳参数组合。我们会尽早停止，以避免过度拟合。另一种是我们的好朋友正则化。我告诉过你，它与线性回归非常相似。在这里，我们还要再看看正则化。不过，让我们先从缓慢学习和提前停止开始。我想我在线性回归的讲座中提到过梯度下降。那是第五讲。我大概简短地说了一下，梯度下降的概念是，我们正在寻找这座小山，以找到最小值，对吧。因此，我们对参数有某种最佳猜测。然后不断迭代，直到目标函数不再下降。所以，我们要找到某种矢量，它能在我们的误差率图中找到微小的变化。我们该如何解释呢？基本上，我们的想法是找到某种组合。例如，这里是第一个组合，然后我们正在寻找一种下山的方法，直到我们达到希望的最佳值，希望的最佳参数组合。所以问题显然是，我们如何找到最佳的前进方向？为此，我们需要计算梯度。为什么要计算梯度？因为它告诉你应该沿着山丘的哪个方向移动，才能使函数最小化。所以，这就是我在这一点上所做的，例如，我会计算出我在这门课程中最后一次画图的机会。期待着。所以，我会计算这一点的梯度，然后我会发现，好吧，这个梯度告诉我，要往这个方向走，然后我会跟着这个梯度，计算下一个梯度，继续往这个方向走。我就这样走啊走，走啊走，走啊走。如果我走得有点远，这个区域的新梯度就会告诉我，不不不，往回走，往回走，往回走。所以，我希望在某一点上，沿着这条梯度线往回走，就能找到最小值，而这正是你要找的。记住，我们要找到误差的最小值。因此，最小值和这里的最小化最小误差是参数的组合。那么我们如何改变参数，使误差最小化呢？是的，我们要做的就是朝着误差最小值的下坡方向前进。你刚才也看到了如果我们的学习速度过快会发生什么。步长有点过大。所以我说，好吧，如果我走得有点远，会发生什么呢？突然间，在我所寻找的山谷的另一侧出现了错误。所以在这种情况下，我跳得有点太远了。所以，你需要找到一个合适的步长，这样你才能找到局部最小值，而不会总是来回跳，跳到你要找的山谷上方。因此，请记住，如果我的步长太小，就会从一边跳到另一边，而无法真正找到这个值。这就是步长过大的危险。另一方面，如果你的步长太小，虽然可以让你在这里往下走，但却要花费很长时间，因为你必须一遍又一遍地计算。所以，要明智地选择这种数值。我总是说要明智选择。这很难做到，不是吗？但在大多数情况下，这需要一点经验和测试不同的值，看看你最终会得到什么结果。所以，如果你有一个步长，然后把它减小一点，突然发现一个更好的最小值，你就知道，好了，你跳过了山谷。还有一点我很幸运从这里开始，不是吗？因为如果我从这里开始，它就会告诉我往另一个方向走。这就是梯度下降法的大问题。它通常只能找到局部最小值，而且取决于你开始时的随机第一个值。那我想说什么？我想说什么？所以呢？这个随机的第一个值。因为问题在于，我们有时只能达到局部最小值。但在现实生活中，我们通常不会只计算一次，而是会取样计算。比如在这里，你可以看到，我们并没有查看所有观测数据的所有参数组合，而是只使用了一小部分观测数据的样本，然后计算梯度步骤。基本上就是这样。通过结合这些不同的步骤或结合数据的不同部分，我们会创造出一种我们称之为随机梯度下降法的东西，它允许你基本上将随机性引入到这一过程中。正如我们多次讨论过的，引入随机性通常会在某种程度上提高优化效果。这里的情况也是如此。因此，随机梯度下降法的最大优势在于，它能帮助你克服局部最小值的问题。我刚刚跳过了几张幻灯片。原因是我们说过前面的计算。所以，基本上是找到误差，找到。找到你的梯度，跟随你的梯度，并在每一步重新计算你的平均值，这并不复杂。问题在于，这需要进行大量的小计算。我把这些漂亮的计算放在这里，是为了向你展示每一个梯度计算实际上是多么简单。我这样做也是想让你看看这个术语。这部分很有意思，因为它向你展示了在每一次计算中，我们都会考虑到残差。因此，我们实际上考虑到了误差，这是有道理的，因为我们正试图将误差率降到最低。因此，微分法在每一次计算中，仍然保留了残差的这一部分。在每一次计算中对于每一个参数这就是文献中所说的反向传播。你可能听说过反向传播 我讨厌这种奇怪的解释 但它在数学上是正确的不过，我不会信口开河。因为我可以把这些都画出来，但别人已经帮我画出来了，所以我会利用他们的解释。嗯，当面这是一段非常非常棒的视频，介绍了反向传播如何在网络中以彩色编码工作，这也是我在这里展示它的原因。我们来看看吧。在这里，我将完全不考虑符号，直接开始分析每个训练实例对权重和偏置的影响。由于代价函数涉及在数以万计的训练实例中平均每个实例的某一代价，因此我们在单步梯度下降时调整权重和偏置的方式也取决于每个实例，或者说，原则上应该如此。但为了提高计算效率，我们稍后会用一个小技巧，让你不需要在每一步中都对每个例子进行调整。这就是我所说的，我们在每一步梯度下降计算中都要考虑到残差。没错他就是这么说的理论上，我们会考虑每个样本和每个步骤，但在计算上，我们并没有这么做。不管是哪种情况，现在我们要做的就是把注意力集中在一个单独的例子上，这个二的图像。这一个训练实例会对权重和偏差的调整产生什么影响呢？比方说，我们现在的网络还没有训练好，所以输出中的激活值看起来很随机。比如 0.5、0.8、0.2。以此类推。记住，这些就是你的概率。所以，我们刚才说的是分类。我们有这些概率。例如，对于 "2 "的例子，你有 0.5 和 1 的概率。因此，你可以看到这些概率并不高。实际上，它并不能估算出这是二的样本。现在，你将看到这个单一训练示例是如何通过网络调整参数的。这就是调整部分，目的是提高一点。每个训练示例都会这样做。现在，我们无法直接改变这些激活。我们只能对权重和偏置施加影响，但跟踪我们希望对输出层进行哪些调整是很有帮助的。由于我们希望它将图像分类为 "2"，因此我们希望第三个值向上推移，而其他所有值向下推移。此外，这些推移的大小应与每个当前值距离目标值的远近成正比。例如，从某种意义上说，提高 2 号神经元的激活度比降低 8 号神经元的激活度更重要，因为后者已经非常接近它应该达到的目标值了。因此，让我们进一步放大，只关注这一个神经元。神经元，我们希望提高它的激活度。请记住，激活度的定义是上一层所有激活度的加权总和，再加上一个偏置，然后将所有激活度输入到类似 sigmoid squish ication 函数或 ReLU 的函数中。因此，有三种不同的途径可以共同帮助提高激活度。你可以增加偏置、增加权重，也可以改变上一层的激活。重点是如何调整权重。请注意权重实际上具有不同程度的影响。与前一层最亮神经元的连接影响最大，因为这些权重乘以较大的激活值。因此，如果增加其中一个权重，实际上会对最终的成本函数产生更大的影响。那么，至少就这个训练实例而言，增加神经元连接的权重就会变得更弱。请记住，当我们谈论梯度下降时，我们并不只关心每个分量是向上还是向下推移。我们关心的是，哪些成分能给你带来最大的收益。顺便说一句，这至少让人想起神经科学中关于神经元生物网络如何学习的理论。海比理论（Hebbian theory）通常被概括为 "神经元一起发火，神经元就一起联结"。在这里，权重的最大增加、连接的最大加强发生在最活跃的神经元和我们希望变得更活跃的神经元之间。从某种意义上说，当看到两个人时，正在发射的神经元与思考时正在发射的神经元之间的联系会变得更加紧密。要说明的是，我确实无法对人工神经元网络的行为是否与生物大脑相似做出这样或那样的表态，而且这种 "一起发射，一起连线 "的想法还带有几个有意义的星号。不过，作为一个非常松散的类比，我确实觉得这很有趣。总之，我们可以通过改变上一层的所有激活状态来提高该神经元激活度的第三种方法是，如果与数字 2 神经元相连的所有正权重神经元都变得更亮，而与负权重神经元相连的所有神经元都变得更暗，那么数字 2 神经元就会变得更加活跃。与权重变化类似，通过寻求与相应权重大小成正比的变化，你将获得最大的收益。当然，我们无法直接影响这些激活。我们只能控制权重和偏置。但就像上一层一样，记下所需的变化是很有帮助的。但请记住，这里放大了一步。这只是第二位输出神经元想要的结果。记住，我们还希望最后一层的所有其他神经元都变得不那么活跃。对于倒数第二层应该发生什么，其他每个输出神经元都有自己的想法。因此，这个数字 2 神经元的愿望与所有其他输出神经元的愿望相加，得出倒数第二层应该发生的变化，同样与相应的权重成比例，并与每个神经元需要改变的程度成比例。通过将所有这些预期效果相加，这就是逆向传播的原理所在。基本上，你可以得到一个列表，列出你希望在倒数第二层产生的效果。一旦有了这些，你就可以递归地将相同的过程应用到决定这些值的相关权重和偏差上。重复我刚才走过的过程，在网络中向后移动。再放大一点，记住这只是一个训练实例希望如何调整每个权重和偏置。如果我们只听从那两张图片的要求，网络最终会被激励将所有图片都归类为 "2"。因此，我们要做的就是对其他每个训练实例进行相同的反向推演，记录每个实例希望如何改变权重和偏置，然后将这些期望的变化平均到一起。这里收集的每个权重和偏置的平均值，粗略地说，就是上一段视频中提到的成本函数的负梯度，或者至少与之成正比。我说 "粗略"，只是因为我还没有对这些 "点拨 "进行精确的量化。但是，如果你理解了我刚才提到的每一个变化，理解了为什么有些变化比其他变化按比例要大，以及它们是如何需要加在一起的，那么你就理解了反向传播实际上在做什么。好了，我希望你能明白为什么我画不出来。我想从中学到的关键是一点。这些视频真的很棒，你可以去看看。另外一点是，反向传播的概念实际上是关于如何。网络结构对优化过程的影响。所以我们可以看到，如果你把它看作是线性回归，那么它就相对简单了，因为在线性回归中，所有的东西都是向前馈送一次。因此，所有的权重、所有的参数都只需调整一次即可。但在一个新的网络中，由于每个节点都是之前发生的线性组合的线性、非线性和非线性转换，突然间，你必须考虑如何将所有这些都考虑在内。因此，从理论上讲，正如我在一开始所说的那样，神经网络是通过它向前输入数据的，至少我们在这里讨论的是神经网络的结构。但随后我们会对神经网络进行调整。我们是在反向优化，因为这些节点之间的关系就是如此。这就是反向传播的基本思想：向前输入数据、调整权重、反向思考。我想，这段视频也让你很好地了解了反向传播在计算上的复杂性。这就是第二个训练实例。这个训练示例在这种情况下，什么是分类？回归任务也是如此。唯一的变化是你要优化的成本函数。其他方面都是一样的。因此，所有这些计算本身都很简单明了，但当它们结合在一起并试图跟踪所有这些计算时，就变得复杂了。好了，这就是梯度下降法的原理，即如何优化参数组合，以最小化成本或其他参数。我们还说过，与回归类似，我们可以用正则化来代替，有几种方法可以做到这一点。我们之前在数字识别的例子中看到，我们有超过 20 万个这样的权重。因此，你可以想象，如果我们真的要优化权重，并将其与数据完美拟合，那么我们的数据就会面临过度拟合的巨大风险。所以神经网络真的很容易出现过度拟合，尤其是如果你在正则化方面不小心的话。因此，与决策树类似，神经网络也有类似的结构，我们对其进行的训练与决策树非常相似。如果你有一个单一的神经网络，而你只是对它进行完美的优化，那么它很可能真的过度拟合了，就像一个单一的决策树一样，但你实际上是在对可能过度拟合的网络进行优化。因此，避免这种情况的一种方法就是使用我们的好朋友拉索正则化（lasso regularisation）。同样，你要记住，对于线性回归来说，所有这些都会反过来困扰你。因此，我们要再次引入某种误差项。所以我们有某种惩罚项。你记得 lambda lambda 是你的正则化参数。选择这个参数，基本上是为了最小化参数的数量。在某种程度上，你也可以使用 lasso 正则化，这和线性回归一样，会导致其中一些参数实际为零。这样，其中一些权重就会对神经网络产生零影响，从而减少网络可能出现的过度拟合。这与线性回归的原理相同。这一点大家应该都知道。唯一不同的是，这里我们要优化的是交叉熵，而不是误差项。除此之外，它的原理和回归是一样的。我想说的另一件事是关于学习，因为这相对来说，我的意思是，可能已经不新鲜了，但仍然非常流行，因为它非常有效。这里的想法是，我们基本上只是随机移除网络的一部分。所以我们说，网络本身真的很复杂。有人在休息时问我，这些节点之间都有联系吗？是的，它们确实相连。例如，你可以看到这个节点与隐藏层中的所有节点都有连接。在某些情况下，停用其中一些节点是有意义的。例如，我们在这里停用了隐藏层中的两个神经元和输入层中的一个神经元。这样，我们仍然可以得到一个很好的估计值，可能不会出现过拟合的情况。所以，你可以把这想成是一种基本的特征选择，不是吗？因此，我们要选择输入向量的哪些部分被输入到我们的网络中。这有点像拉索正则化，但又不完全像拉索正则化，但我们基本上是在去掉其中的一些部分，去掉其中的一些特征组合。因此，我们在移动复杂性。就像修剪决策树一样。我们正在移除神经网络的一部分。是的，这些幸存的单元将承担那些丢失单元的工作。因此，在这种情况下，我们会增加它们的权重来进行补偿。同样，这也是因为它们必须承担更多的工作。我认为这也与大脑的工作原理密切相关。所以我不知道你是否听说过，比如在一些脑部受伤的人身上，他们大脑的其他部分会承担受伤部分曾经承担的工作。这也是类似的原理。这些幸存的单元代替了被移除的单元。是的，这可以防止新的单元变得过于专业化。我们前面已经举过这个例子。如果你的训练数据不平衡，情况尤其如此。同样，不平衡的训练数据会导致你的神经网络本身变得过于专业化。我见过这种情况。我的一位同事就遇到过这种情况，他对自己的神经网络感到非常沮丧。他的神经网络非常沮丧，因为它总是在预测同类。不，尽管他一直在为二元分类任务寻找 "是 "和 "否 "的良好组合。而解决这个问题的简单方法就是重新平衡训练数据。所以，回到最基本的地方，看看你的预处理，再试一次。这通常会有所帮助。作为总结，我还想提的一点是，我们看到网络调整本身听起来简单，但其实很复杂。我们可以选择隐藏层的数量，也可以选择每层的单元数量。之前有人问我，我的多层网络（我们这里的多层网络）的节点数和节点数都少于训练实例，而且下一个隐藏层的节点数也少于我的单层网络，这是否是有意为之？答案是肯定的，也是否定的。所以，这是系列讲座的目的，但并不总是这样。之所以会这样做，是因为你基本上有两种方法可以将复杂性引入模型。你可以增加每层神经元的节点数，或者增加层数，这也会增加节点数。因此，这两种方法都完全有效。这两种方法都很有效，至于哪种方法更有效，通常取决于你的问题。因此，并没有一种方法能真正决定这一点。现代的想法是，我们可以在每层设置相当多的单元，因为我们现在正处于计算效率较高的阶段。然后，你可以通过正则化来控制过度拟合，例如，剔除学习（dropout learning），这样只需停用其中一些单元，就能获得不错的结果。另一种方法是增加层数，这也是深度学习背后的思维过程。因此，有些人认为，对于非常复杂的任务来说，增加没有那么多节点的多图层效果更好。究竟哪种方法更有效，基本上取决于非线性和复杂性的大小。你必须自己试一试。取决于你的问题。两者都有效。两者在计算上是相似的。所以隐藏层数、每层单元数。然后你必须选择正则化调整参数。例如，你的丢弃率（dropout rate），基本上就是要停用多少个节点，或者你的正则化拉索（lasso）的强度。所有这些，你都必须为每一层单独设置。因此，你必须为梯度下降或随机梯度下降做出很多决定，这是避免局部最优化的首选方法，包括学习率、批量大小以及实际运行的次数。因此，所有这些参数都会增加复杂性。这还不算参数估计，我们刚刚已经看到参数估计非常复杂。所以，很多选择、很多估计、复杂的模型。但我希望你们能同意。但这个想法相对简单。所以说，有多个层级来向前输入数据是一个相对简单的想法。只是欠拟合和调整让它变得如此复杂。我也希望我听起来不会太消极。过去我曾听说有人抱怨我对神经网络过于消极，因为我认为它们非常专业化，但我希望我给了你一个很好的概念，让你知道它们确实适用于哪些任务。但我希望我的介绍能让你对神经网络有一个很好的了解，它们确实能很好地完成一些任务，只是有时候用得太多了，因为你需要大量的数据。好了，以上就是我们对神经网络的简要论述。众所周知，这是最后一讲。所以，下周别来这里听移动人工智能了。我下周不会来了。我明天会在你们另一栋楼的计算机实验室里。我们会研究如何实现神经网络小组作业的截止日期也快到了。我知道你们现在不想听到截止日期。我可以想象你们已经厌倦了这些。但请记住这一点。请在考试前仔细检查考试时间和地点。如果有任何更新，您应该会收到一封电子邮件，但不要冒险。在去考场之前，请仔细查看网站。好了，我在这里又添加了一个考试网站的链接。现在我想做的是一起简单了解一下考试情况，因为我认为现在是了解考试信息的最佳时机。我把链接加到哪里了？在这里你们还有其他考试吗？只有这个就这一次你真幸运这将是你第一次看到试卷显然也是本学期最后一次考试 很好这就是它的样子这是扫描件显然时间和日期略有不同我怎么强调都不为过。请仔细阅读。你不会相信每年有多少人没有仔细阅读说明，当这种情况发生在你身上时，你会非常沮丧，因为不仔细阅读很容易做错事，然后你就会因为紧张而丢分，所以你会紧张。慢慢来。你有两个小时的时间，足够你答出一个非常好的答案。请相信这一点。例如，这里会告诉你回答第一部分的所有问题，并回答第二部分中的一个问题。所以，这次你又会遇到一道选择题。或者说，多选题部分将是第一部分。然后第二部分会有一个开放部分。因此，今年的结构是一样的。问题显然会有所不同。我希望你们知道这一点。对于选择题作弊，你会得到一本额外的小册子，你必须在上面填写。因为它们是机读的。因此，请仔细阅读如何填写，以便计算机和机器能够读取你的答案。这样我们就不用让教学管理人员去解读你的选择了。好的。我们利用他们的工作量。他们在 12 月份工作得很辛苦。这就是选择题表。你会有一些小册子。然后你必须清楚地标出你的答案。你必须用铅笔。我们会提供给你。这样机器才能正确读取。是的，非常细的标记可能无法读取。这就是为什么要提供一支软铅笔。等等等等。答对一项给 1 分。一个答案不会被扣分。所以，即使你不完全确定，也请尝试一下。不给分也不扣分。如果没有尝试，或者某一行有多个选项。这是因为所有这些选择题都只有一个正确答案。因此，试卷会是这样的尝试所有问题。您总共有 35 分。每题答对可得 5 分。这些是你会遇到的一些例子。因此，题目中会有一些类似于选项中线性回归问题的例子。然后你会有一张单独的纸，并清楚地标出是哪一个。谁想回答问题一？你能看到吗？你能做到吗？线性回归问题的例子是什么？将机器零件分为有故障和无故障。预测一个人是否会违约，根据行为细分客户群，或预测音乐会的入场人数。正是如此。我们在预测一个数字。然后你会非常清楚地标出这个数字。有人说最后一个会持续吗？好吧没关系好的，一共有7道题，35分第二部分是开放式讨论题我再次用粗体字标出请回答1或2中的一个。我之所以用黑体标出，是因为有些人试图同时回答这两个问题，这样显然就没有足够的时间来阐述问题，所以请仔细阅读。好的，今年在这里你可以选择逻辑回归。决策树这两种方法，对它们进行描述、比较和讨论。或者建立一个训练机器。学习模型有陷阱。解释如何避免过度拟合。谈谈重采样。谈谈正则化，这样你就能看到这些分数是怎么来的。然后在这里划分，例如 35 分和 30 分。所以你总共能得到 65 分。有人问我，答案应该写多少？这其实没有什么规定。如果有道理的话，你应该回答得足够多。简明扼要的答案就可以了，这样我就不用读起来像读书报告之类的东西。只要你在所有方框内打勾，正确简洁地回答问题。我对此很满意。考虑一下时间你总共有两个小时。你可以估算一下，两个小时可以得到 100 分。那么一小时 50 分，就像估算每道题或每个部分大致应该花多少时间一样。在此基础上，我希望你们都在练字。我必须读一下。请确保字迹清晰。否则，我不能给你们分数。好吗？如果我看不懂，你就得不到分数。还有一点，我想让你们看一下我在这里使用的描述、比较和讨论的关键词。想想这些词的意思。我所要求的 "描述 "是指实际描述方法的作用。这就是描述。你可以使用公式，但并非必须。你可以使用数字示例，但不必如此。你可以用自己的话来描述这些方法。这样既合理，又简洁正确。比较和讨论，然后是批判性分析部分。因此，我希望你们思考这些方法，不只是它们做了什么，而是它们是如何做的。这两种方法有什么不同？你什么时候会使用其中一种方法？你能举些什么例子？可以。比如，什么时候逻辑回归比决策树更好？等等。我说用例子。如果我这么说了，请举例说明。请不要忽视这一点。这些就是这个问题的例子。在这里，我们解释了关键词。这里我们讨论了。所以我们又有了描述性的部分。然后是关键部分。所以，我希望这一点相对清晰。论文到此结束。我们会给你们一些小册子。如果你需要更多的纸，我想你可以再要一些，但不要太多，好吗。是的，你也可以在上面画画。图片可以让我在圣诞假期好好看看。好的对考试有什么问题吗？有对不对？或者它只是。我想应该是内衬是的，嗯？我去看看播音员哦，天哪，是的，那一定很有趣不，我觉得那条线 我是说，老实说，在我看来 它们应该是方形的 但我觉得你会有内衬的好吧 还有问题吗 Yeah.还有其他问题吗？你在课程一开始就很害怕考试记得吗？你现在感觉好点了吗？是的，没有不，哦，天哪。另一件你可以考虑的事情是，计算机实验室为你的课业做好了准备。基本上，我在实施过程中提到了理论，希望能更好或更深入地解释方法。但基本上，讲座会为考试提供素材，然后计算机实验室会为课业提供素材。因此，你可以这样来思考这两个元素是如何协同工作的。还有一点，你可以再看看课程的学习目标，因为它们真正解释了我是如何构建考试的，因为这就是我想要测试的。我这样做不是为了好玩。所以，我不是为了好玩而考试。我写考试，是因为我想看看你们是否达到了课程的学习目标。这是评估你是否达到目标的正式方式。你是否达到了课程的终点。有道理吗？有道理还有其他问题吗？关于小组作业的问题还有其他问题吗？你们还有十分钟时间向我提问然后我就永远离开了不，我仍然可以通过电子邮件联系你们如果还有什么问题，给我发邮件。我很乐意和你谈谈如果你有关于考试或课程的问题，或者你想仔细检查我们在复习过程中涉及的一些材料。请给我发电子邮件，我很乐意提供帮助。还有其他问题吗？没有了，我很高兴。你很开心吗？压力很大，但很开心好吧，我，我不能改变这一点。不幸的是，我会，我会，如果，我可以但你会熬过去的我觉得我们在一起很开心所以我希望你能喜欢这些讲座 并觉得它们有用和有趣。我会在你们考试前来看你们，因为我会在那里送试卷，然后在考试后收试卷。如果你们不满意，可以在考试后直接向我投诉，但那时已经太晚了。就这样吧。好了，如果我没看到你，那是因为我太快了还是你太快了。如果你正在庆祝，祝你节日快乐，新年快乐，很高兴教你，希望以后还能再见到你。非常感谢。谢谢