第5章

视频讲解

GAN-FinWoBERT：对抗训练的中文金融预训练模型

5.1 对抗训练目的

BERT样式的预训练语言模型的评估一般是通过执行下游任务来进行评测的，评测过程中使用成百上千的样本数据。在许多真实场景中，获取高质量的标注数据是昂贵和耗时的，而描述目标任务的未标注数据通常可以很容易地收集到。基于半监督生成对抗网络生成一些对抗样本，在模型训练的时候采用对抗训练的方法，对词嵌入添加扰动，可以提高模型应对对抗样本的鲁棒性；同时可以作为一种正则化(regularization)，减少过拟合，提高泛化能力。

本章对第4章的自建未标注语料库经过生成对抗网络，在FinWoBERT模型的预训练阶段，增加对抗样本；也就是说，生成器产生类似于数据分布的伪样本，而WoBERT被用作一个判别器。通过这种方式，既利用WoBERT产生高质量的输入文本表征的能力，也利用未标注的语料库来帮助网络在最终任务中推广其表征。采用对抗训练的方法，训练模型GAN-FinWoBERT，并执行下游情绪分类任务，与FinWoBERT模型的评测结果进行比较，希望通过对抗训练的改进使模型达到更高基准水平。

5.2 对抗训练原理

生成对抗网络(Generative Adversarial Networks or Generative Adversarial Nets，GAN)同时训练两个网络模型：一个生成网络模型(generative model)捕捉样本分布生成伪样本欺骗判别网络模型(discriminative model)，另一个判别网络模型(discriminative model)进行二分类预测，判断是不是真实样本，估计来自真实(训练)样

本的概率。生成网络和判别网络都是反向传播神经网络。生成器(generator)的训练目标是尽可能迷惑判别器(discriminator)，使判别器犯错的概率最大化，让其无法判断一个样本是来自训练样本还是生成网络模型产生的，不存在两方共赢的局面；而判别器要尽可能区分生成器生成的对抗(伪)样本和训练(真实)样本。生成对抗网络将生成器和判别器视为一个整体，对于一个生成样本(generative neural sample)，判别器的评分高，说明生成器伪造能力很强，判别器的评分低，说明可以有效区别真伪。生成对抗网络是一种特殊的数据扩增方法，是不太可能自然发生的输入；而其他的数据扩增方法一般是使用转换方式或合成方式来扩充数据。生成对抗网络与LSTM模型作者另一篇论文中描述的可预测性最小化(predictability minimization)算法的思路相似：基于两种对立的力量，每个代表性单位都有一个自适应预测器，试图从剩余的单元中预测该单元，每个单元都试图对环境做出反应，使其可预测性最小化；这鼓励每个单元从环境输入中过滤出“抽象概念”，这样这些概念在统计上就独立于其他单元所关注的概念；可预测性最小化算法既可以消除线性模型，也可以消除非线性模型的输出冗余(即过拟合)。

神经网络容易受到对抗样本攻击(adversarial example attack)，softmax分类器容易受到对抗样本的影响，对伪样本进行了错误分类，即使是在测试集上获得出色性能的分类器，也无法学习确定正确输出标签的真实基础概念。生成对抗网络通过训练对抗样本和真实样本的混合，可以对神经网络进行一定程度的正则化，从而成功地训练非线性模型更强大。

生成对抗网络训练过程实际上是求解纳什均衡的最优解。纳什均衡(Nash equilibrium)是指博弈中这样的局面，对于每个参与者来说，只要其他人不改变策略，他就无法改善自己的状况。对应地，对于生成对抗网络，情况就是生成器恢复了训练数据的分布(造出了和真实数据一模一样的样本)，判别模型再也判别不出来结果，准确率为50%，约等于乱猜。这时双方网络都得到利益最大化，不再改变自己的策略，也就是不再更新自己的权重。

生成对抗网络的对抗博弈可以通过判别函数 $D(X)$：$\mathbb{R}^n \to [0,1]$ 和生成函数 G：$\mathbb{R}^d \to \mathbb{R}^n$ 之间的目标函数的极大极小值来进行数学化的表示。生成器 G 将随机样本 $z \in \mathbb{R}^d$ 分布 γ 转换为生成样本 $G(z)$。判别器 D 试图将它们与来自分布 μ 的训练样本区分开来，而 G 试图使生成的样本在分布上与训练样本相似。对抗的目标损失函数的数学表示式为：

$$V(D,G):=\mathbb{E}_{x\sim\mu}[\log D(X)]+\mathbb{E}_{z\sim\gamma}[\log(1-D(G(z)))]$$

式中，$\mathbb{E}$ 表示关于下标中指定分布的期望值。生成对抗网络解决的极小极大值的数学表示式为：

$$\min_G\max_D V(D,G):=\min_G\max_D\{\mathbb{E}_{x\sim\mu}[\log D(X)]+\mathbb{E}_{z\sim\gamma}[\log(1-D(G(z)))]\}$$

对于给定的生成器 G，$\max_D V(D,G)$ 优化判别器 D 以区分生成的样本 $G(z)$，其原理是尝试将高值分配给来自分布 μ 的真实样本，并将低值分配给生成的样本 $G(z)$。反过来说，对于给定的判别器 D，$\min_G V(D,G)$ 优化 G，使得生成的样本 $G(z)$ 将试图“愚弄”判别器 D 以分配高值。

如果使用最大似然估计(maximum likelihood estimation)方法求解上面这个最优化问题,难以逼近分布函数中的 θ 参数。无论是训练样本还是生成样本,生成对抗网络都不需要任何马尔可夫链或展开的近似推理网络,不需要明确地定义概率分布,而是训练生成机器从期望的分布中抽取样本,经过生成器得到输出伪样本,这些输出集合就是生成样本分布,优化目标是生成样本和训练样本两个分布的差异程度 θ_G 的最小化,这个最小值无法直接计算,但可以在假设生成样本分布变化足够缓慢、判别器输出两个分布的差异程度最大值 θ_D 保持在最优解附近的情况下,通过提升其随机梯度来更新判别器的 $\nabla\theta_D$、降低随机梯度来更新生成器的 $\nabla\theta_G$,逐步逼近得到两个分布间差异的最小值。从纯数学的角度来看,生成对抗网络的最优解求解过程并不严谨,然而深度神经网络的数值计算方法为解决生成对抗网络极小极大问题提供了一个实用的框架。

5.3 对抗训练实现方法

对抗样本数是为了混淆神经网络而产生的特殊输入,会导致模型对给定输入进行错误分类。按照攻击环境的不同,攻击可以分为黑盒攻击、白盒攻击或者灰盒攻击。黑盒攻击是指攻击者对攻击的模型的内部结构、训练参数、防御方法(如果加入了防御手段的话)等一无所知,只能通过输出与模型进行交互。白盒攻击与黑盒攻击相反,攻击者对模型的一切都可以掌握。灰盒攻击是介于黑盒攻击和白盒攻击之间,仅了解模型的一部分(例如,仅拿到模型的输出概率,或者只知道模型结构,但不知道参数)。

通过对生成样本进行定性和定量评估,基于快速梯度符号法(Fast Gradient Sign Method,FGSM)的对抗目标函数训练是一种有效的正则化器,使用这种方法来训练可以有效降低错误率。对抗样本快速梯度符号法是一种白盒攻击,其目标是确保分类错误,目前大多数攻击算法都是白盒攻击。对抗样本快速梯度符号法的工作原理是利用神经网络的梯度来创建对抗样本,该方法使用相对于输入样本的损失梯度来创建使损失函数最大化的新样本。这样做是因为其目标是创造一个最大化损失的样本。实现这一点的方法是找出样本数据集中每个样本对损失值的贡献程度,并相应地添加一个扰动(使用链式规则去计算梯度可以很容易地找到每个输入样本的贡献程度)。此外,由于模型不再被训练(因此梯度不针对可训练变量,即模型参数),因此模型参数保持不变。唯一的目的就是使一个已经受过训练的模型发生错误的分类。对抗样本快速梯度符号法的数学表示式为:

$$\text{adv\_x} = x + \in \times \text{sign}(\nabla_x J(\theta, x, y))$$

其中,adv_x 表示对抗样本(即生成样本),x 表示原始输入样本,y 表示原始输入标签,$\in$ 表示噪声的干扰程度(乘法器以确保扰动很小),θ 表示模型参数,J 表示损失函数。

具体的操作步骤是先对 FinWoBERT 进行对抗训练,再对训练好的 GAN-FinWoBERT 模型执行情绪分类任务进行评估。

5.4 定义 GAN-FinWoBERT 模型

GAN-FinWoBERT 模型的工作流程如图 5.1 所示,其中,生成器采用多层感知机

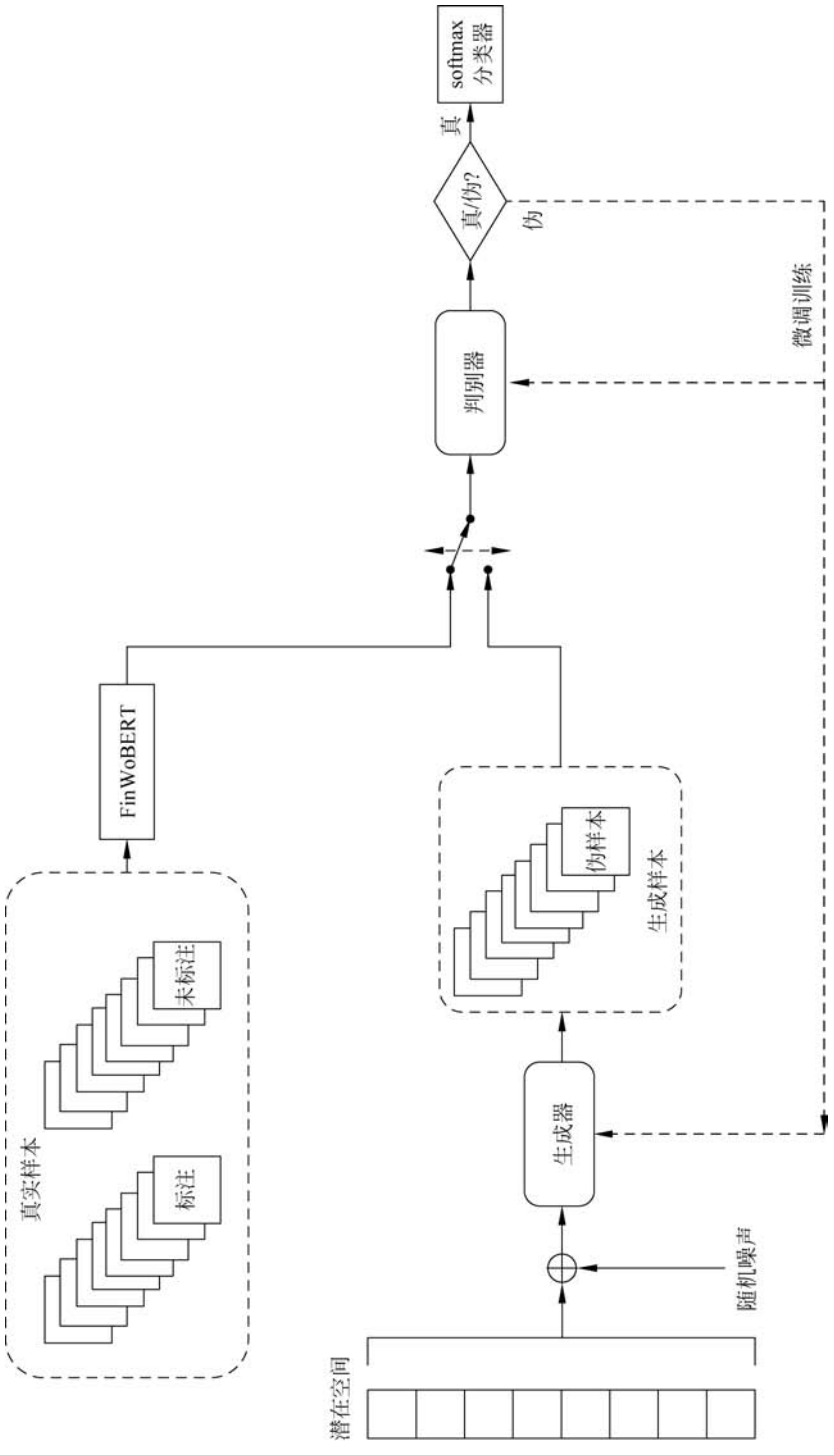

图 5.1　GAN-FinWoBERT 模型工作流程

（Multi-Layer Perceptron，MLP）神经网络，它的输入是一个 100 维的白噪声序列，服从正态分布 $N(0,\sigma^2)$，它输出的生成样本是向量。判别器也采用多层感知器，生成样本连同由 FinWoBERT 计算的未标注和标注的语言表征向量，作为判别器的输入。

将生成对抗网络应用于半监督分类任务时，只需要对最初的生成对抗网络的结构稍微改动，即把判别器模型的输出层替换成 softmax 激活函数分类器。假设训练数据有 c 类，那么在训练生成对抗网络模型的时候，可以把生成模拟出来的样本归为第 $c+1$ 类，而 softmax 分类器也增加一个输出神经元，用于表示判别器模型的输入为“假数据”的概率，这里的“假数据”具体指生成器生成的样本。因为该模型可以利用有标签的训练样本，也可以从无标签的生成数据中学习，所以称之为“半监督”分类。

对抗训练完成后，屏蔽生成器部分程序，保留原始训练模型的其余部分执行下游任务，就得到 FinWoBERT 的对抗训练模型——GAN-FinWoBERT 模型。

5.5 建立 GAN-FinWoBERT 模型

谷歌 TensorFlow 官方网站和 GitHub 账号网页提供了对抗样本快速梯度符号法（adversarial example using FGSM）的程序代码样例，本章项目的部分代码是根据样例的代码修改的。

预训练模型的对抗训练模型的一般建立过程包括：导入需要的软件库，加载预训练模型，数据预处理，将样本输入模型并得到概率最高的分类结果，计算与输入样本的损失梯度，添加噪声，定义生成函数，将生成样本输入判别器进行判断。

在本项目中，第 4 章已经完成了加载预训练模型、数据预处理、将样本输入模型并得到概率最高的分类结果，因此这三步可以省略。

5.6 训练 GAN-FinWoBERT 模型

模型训练过程主要使用了正则化技术，可以改善或者减少过度拟合问题。过拟合也就是高方差问题，说明模型可能太过庞大、变量太多；当只有少量数据集来进行模型训练，约束这个变量过多的模型，那么就会发生过拟合。解决方法一是尽量减少选取变量的数量，另一个就是正则化。正则化保留所有的特征变量，但是减少特征变量数量级（magnitude），即模型参数 θ_i 数值的大小。正则化方法非常有效，尤其是当模型有很多特征变量而每一个变量都能对预测产生一点儿影响时。

谷歌的丢弃（dropout）算法是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其从网络中暂时丢弃。随机梯度下降中是随机丢弃，因此每个批次的批处理都在训练不同的网络。增加了丢弃算法后，神经网络的训练和预测就会发生一些变化。对抗性训练可以提供丢弃算法以外的正则化收益。通用的正则化策略（例如，dropout、预训练和模型平均）并不能显著降低模型对付对抗样本的脆弱性，但改用非线性模型可以做到。使用快速梯度符号法来训练生成对抗网络也通过丢弃算法进

行正则化的最大输出网络(maxout network)，maxout 能够降低对抗训练判别器的错误率。

5.7　对抗训练项目结论

与第 4 章 FinWoBERT 模型评估相同，GAN-FinWoBERT 模型依然使用第 3 章自建标注金融情绪语料库，进行模型评估，评估指标见第 3 章。

GAN-FinWoBERT 模型的最佳准确度为 0.909 07，对应的损失值为 0.014 16。

在相同评测数据的中文金融情绪分类任务下，GAN-FinWoBERT 模型的准确度(0.909 07)比 FinWoBERT 模型(0.887 37)提升了 0.0217，比 $\text{BERT}_{\text{BASE}}$ Chinese 模型(0.839 59)提升了 0.0069 48，证明对抗训练方法有效地提升了预训练语言模型。

GAN-FinWoBERT 系统地提高了这种体系结构的健壮性，同时不会给推断带来额外的成本。

小结

为了实现基于对抗性的领域适应，本章项目使用生成对抗网络，生成器学习区分源和目标域的特征，判别器帮助生成器产生的特征对于源和目标领域是不可区分的。这里的生成器是简单的特征提取器。对抗训练是增强神经网络鲁棒性的重要方式，提供了一种正则化监督学习算法的方法。对于生成对抗网络所包含的两个网络模型，生成网络只用于对抗训练，而在使用 GAN-FinWoBERT 执行下游任务时只需要判别器。由于添加了一个可学习的判别器，学习特定于领域的数据集特征提取，可以帮助区分源和目标域，从而帮助预训练语言模型产生更鲁棒的特征。

数据增强是一种有效缓解数据稀疏问题的方法，对抗训练直接来源于使用对抗样本进行数据扩增，但是当偏好的强特征比竞争的弱特征更难提取时，数据扩增的效果就会降低。采用最大似然估计的序列生成模型在生成采样过程中遇到暴露偏差问题，生成对抗网络是一种有效缓解暴露偏差问题的训练策略。

快速梯度符号法(FGSM)沿着梯度方向对非线性模型仅添加一次线性扰动生成攻击样本，简单有效、速度较快，但是非线性模型可能在极小范围内剧烈变化，单次梯度更新步长过大的话不一定攻击成功。FGSM 虽然计算梯度，但不更新参数，相当于在每个梯度方向上都走相同的一步，而快速梯度法(Fast Gradient Method，FGM)则是根据具体梯度通过参数测量出扰动，FGSM 和 FGM 都仅做一次迭代。投影梯度下降(Project Gradient Descent，PGD)则是多次迭代找到最优扰动，每次迭代都会将扰动投射到规定范围内；但是 PGD 计算耗时成本高，极大影响效率，不精确 PGD、释放对抗训练(Free Adversarial Training，FreeAT)、仅传播一次(You Only Propagate Once，YOPO)、释放大批量(Free Large-Batch，FreeLB)、快速梯度投影法(Fast Gradient Projection Method，FGPM)等算法通过同时利用更多信息加快训练速度。除了 FGSM，本章项目还可以使用其他算法。