技术领域

本方法涉及可解释机器学习与表征学习领域，尤其是基于概率视角的可视化神经网络输入对决策影响的归因方法和分析神经网络表征的分析方法。

技术背景

为了解释神经网络这一黑盒，归因方法成为活跃的研究领，该领域尤其要解释模型输入对输出的影响。Gradient Maps[1] 和 Saliency Maps[2]基于输出神经元对输入特征的梯度。Integrated Gradient[3] 和 SmoothGrad[4] 通过平均化多个输入的来提升基于梯度的方法。LRP[5]，DTD[6]，GuideBP[7]修改了反向传播规则，除了基于梯度，基于扰动的方法不需要反向传播，把模型当作黑盒。Occlusion[8] 通过将图像的 patch 替换成 0 来测量分类性能的下降。Per-Sample 和 Readout[9]通过限制信息流来保留必要信息。

表征学习是非常活跃的广阔研究领域，有大量的方法来学习表征，经典的比如说独立主成分析 ICA[10]，SOM[11]等方法，近一些的比如 word2vec[16]等方法，现在的一些生成模型如 vaes[14]，语言模型 berts[15]，自监督学习方法 DIM[17]等。

互信息估计在机器学习领域已经有很长历史了，且广泛的用于各种学习任务，互信息不能直接计算，需要近似，近年来随着神经网络的发展，基于这一函数近似器，互信息估计迎来了新的发展，InforNCE[13], JSD 散度[12]等互信息的计算方法不断提出，改进。一般的，基于互信息的上界的优化可称为信息瓶颈，信息瓶颈可广泛的用于分析学习过程，增加模型鲁棒性等。

基于互信息下界的优化可称为信息最大化原则，可广泛用于各类自监督学习任务，如分类，分割，聚类等。

方法内容

本专利方法了一种解释和分析神经网络决策和表征的框架。

本方法基于局部表征与全局表征间的互信息估计。本方法创新之处在于将决策解释和表征分析相融合，特别适用于分析各类神经网络。

1. 互信息为测量随机变量之间依赖程度的度量，使用互信息测量表征的优势在于其概率解释和维度无关，劣势在于难以计算，需要充满挑战的近似方法。一般的如果是两个随机变量 X 和 Y（该方法仅涉及两个随机变量），互信息可定义为 $\mathrm{I(X,Y)} = D_{KL}(\mathrm{p(x,y)||p(x)p(y)})$，为了近似方便可写为$D_{KL}(\mathrm{p(x|y)||p(x)})$

2. 我们构造互信息最大化模块，用该模块来分析表征，我们使用 InforNCE 下界来估计神经网络局部表征和全局表征的互信息并优化。令 x,y 分别代表局部表征和全局表征，N 为样本总数，K 为局部表征数目，x_{ij} 代表第 i 个样本第 j 个局部表征，y_i 代表第 i 个样本的全局表征.则

$$\mathrm{I(x,y)} \geq \mathrm{E}\left[\frac{1}{NK}\sum_{I=1}^{N}\sum_{J=1}^{K} log\frac{e^{f(x_{ij,}\,y_i)}}{\sum_{m=1}^{N}\sum_{k=1}^{K} e^{f(x_{mk,}\,y_m)}}\right]$$

有了局部表征和全局表征的互信息最大化模块，我们就可以分析神经网络的表征，特别是全局表征。

3. 接着我们构造信息瓶颈模块，用该模块提取出全局表征中的决策充分互信息,用来解释决策以及结合信息最大化模块来分析表征，我们使用 VIB 上界来估计神经网络局部表征和全局表征的互信息并优化，令 r(y_i)表示对 p(y_i)的近似，则 VIB 上界可表示为

$$\mathrm{I(x,y)} \leq \mathrm{E}\left[\frac{1}{NK}\sum_{I=1}^{N}\sum_{J=1}^{K} D_{kl}(\mathrm{p}(y_i|x_{ij})||\mathrm{r}(y_i))\right]$$

4. 为了不使两个模块影响对方，我们在局部表征或全局表征上添加噪声来实现信息瓶颈，令噪声 e~(0，I)，则在局部表征添加噪声可表达为$x_{ij}^{'}$ = αe+(1-α)$x_{ij,}$，其中系数α，$0 \leq \alpha_{pq} \leqslant 1$ 既可以手工设置,也可以由神经网络推断出。该系数控制噪声添加的程度，当它等于 1 时$x_{ij,}$完全变为噪声，当它为 0 时$x_{ij,}$不添加任何噪声。当我们添加完噪声后，我们可以使用加入噪声后的表征$x_{ij}^{'}$来参与信息瓶颈的计算。使用未添加噪声的表征x_{ij}来参与互信息最大化模块的计算，二者不再干扰。
5. 最后我们将两个模块放入到需要分析的模型中，形成最终的目标函数。令 f(x_i)代表原始需要分析模型的目标函数，则我们放入上述模块后形成的最终目标函数 L 为 $\mathrm{f}(x_i) - \beta_1\mathrm{E}\left[\frac{1}{NK}\sum_{I=1}^{N}\sum_{J=1}^{K} log\frac{e^{f(x_{ij,}\,y_i)}}{\sum_{m=1}^{N}\sum_{k=1}^{K} e^{f(x_{mk,}\,y_m)}}\right] + \beta_2\mathrm{E}\left[\frac{1}{NK}\sum_{I=1}^{N}\sum_{J=1}^{K} D_{kl}(\mathrm{p}(y_i|x_{ij}^{'})||\mathrm{r}(y_i))\right]$，其中β为超参数.
6. 我们结合模型图图 1 来说明上述模块的具体实现。
   推断网络（infer network）可以由多层卷积或多层 mlp 实现，以被分析模型（Base Network）第一次前向传播后预先选取的几层的输出为输入，我们把这几层的输出经过连接，尺寸调整后放入推断网络，推断网络最终输出系数α，$0 \leq \alpha_{pq} \leqslant 1$

互信息最大化估计器(Informax estimator)可以由单层或多层 mlp 实现，以原始局部表征$x_{ij,}$和全局表征 y_i为输入，估计器最终输出互信息 $\mathrm{I}(x_{ij,},\mathrm{yi})$下界。

信息瓶颈层（Infor bottleneck）[20]可以由单层或多层 mlp 实现，该层有两个功能，第一以原始局部表征$x_{ij,}$为输入,结合系数α和噪声 e 来生成局部表征$x_{ij}^{'}$，第二是以$x_{ij}^{'}$和 y 为输入，计算互信息 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$上界 。

我们现在把这些模块整合起来来看看怎么进行解释决策和分析表征。给定输入数据和预训练的被分析模型，我们把上述模块插入 Base Network 中，然后进行第一次前向传播，得到系数α，然后进行第二次前向传播，再利用目标函数 L 进行反向传播，最终互信息最大化估计器和信息瓶颈层输出的近似互信息 $\mathrm{I}(x_{ij,},\mathrm{yi})$和 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$就可以用于解释决策，分析表征。

具体实施方式

本方法适用范围广阔，为了不失一般性，下面选取基础的分类问题的作为实施案例。

1.神经网络分类解释

不失一般性我们使用 resnet50[19]作为 Base Network，我们使用 stl10 数据集来训练模型，我们首先在 stl10 数据集上训练被分析模型 resnet50，然后我们判断该问题在架构上属于 Base Network + output 类型，我们选取 resnet50 的 layer1 block 中的最后一层作为局部表征，fc 层作为全局表征，每个 block 中的最后一层作为 infer Network 的输入，接着微调插入上述模块的模型得到互信息 $\mathrm{I}(x_{ij,},\mathrm{yi})$和 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$，接着我们便可以可视化出来，效果见图 2，图 3。我们的方法可以得到输入对输出的影响程度，这一影响程度使用互信息来度量，影响高的部分在热力图上呈现红色。图 2，图 3 展示了 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$的热力图和原始图的结合，$\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$的热力图，原始图，在热力图中，红色部分代表互信息高即影响程度高，这解释了模型的决策依据。

2.小样本学习解释

不失一般性我们使用原型网络（protonet）[18]作为 Base Network，我们使用 cub 数据集来训练模型，我们首先在 cub 数据集上训练被分析模型 protonet，然后我们判断该问题在架构上属于 BaseNetwork + downstream network 类型，我们以 resnet50 为 protonet 的特征提取器，为了可视化方便我们设置学习方式为 5way1shot，接着选取 resnet50 的 layer1 block 中的最后一层作为局部表征，fc 层作为全局表征，每个 block 中的最后一层作为 infer Network 的输入，接着微调插入上述模块的模型得到互信息 $\mathrm{I}(x_{ij,},\mathrm{yi})$和 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$，最后我们便可以可视化出来。图 3 为 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$的热力图和原图的结合。图 4 为 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$的热力图。图 5 左图为输入的图像。我们可以看到图 3 和图 4 解释了原型网络的决策，它关注于鸟类身上的典型部位（嘴，眼睛，特殊羽毛等）并以此为判断依据。

3.小样本学习表征分析

尽管网络中各层输出都可视为表征，但是由于计算复杂度过大，我们不得不只关注重要表征，特别是模型的中间层，靠近输出的层，这些具有分析价值，所以我们接下来分析 fc 层，我们接着利用上述方法构造训练原型网络，然后我们来看看怎么分析。我们继续可视化，图 5 右图展示了 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$的热力图和原图的结合，这代表着 fc 表征中包含的对输入的一般互信息，即 fc 中包含了多少输入的信息，图 6 右图展示了 $\mathrm{I}(x_{ij,},\mathrm{yi})$的热力图和原图的结合，这代表着 fc 表征包含的对输入的充分互信息，即 fc 中包含了多少使得决策充分的互信息，可以观察到红色区域是显著小于图 5 的，这代表着 fc 中有许多使得决策冗余的输入信息，图 6 左图展示了 $\mathrm{I}(x_{ij,},\mathrm{yi})$和 $\mathrm{I}\left(x_{ij}^{'},\mathrm{yi}\right)$差值的热力图和原图的结合，代表着上述一般互信息和充分互信息的差值。我们可以试着用这些来分析学习过程和不同的学习方式。

附录

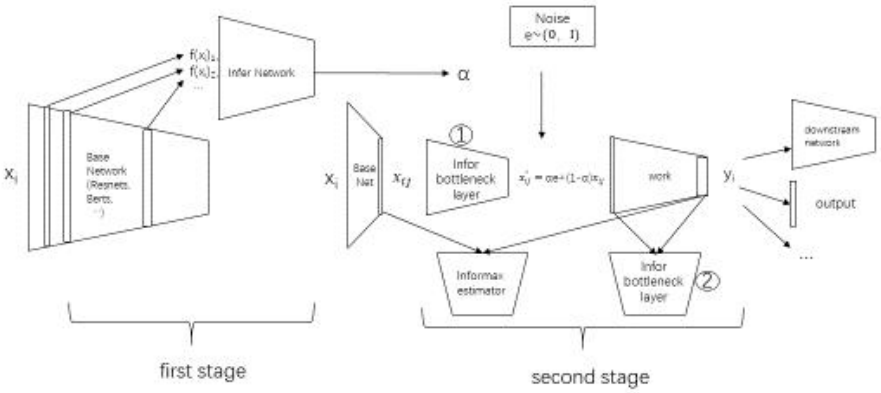
Noise
e~(0, 1)
Infor Network
α
Base Network (Resnets, Berts, ...)
Base Net
Infor bottleneck layer
①
work
downstream network
output
...
Informax estimator
Infor bottleneck layer
②
first stage
second stage

图 1

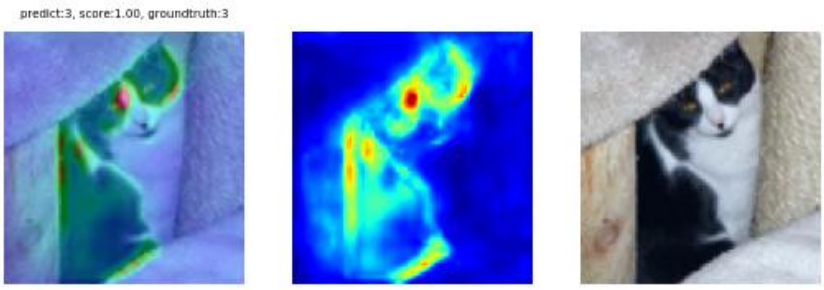
predict:3, score:1.00, groundtruth:3

图 2

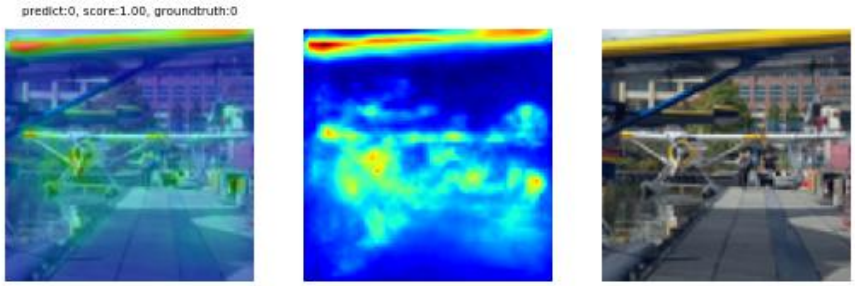
predict:0, score:1.00, groundtruth:0

图 3

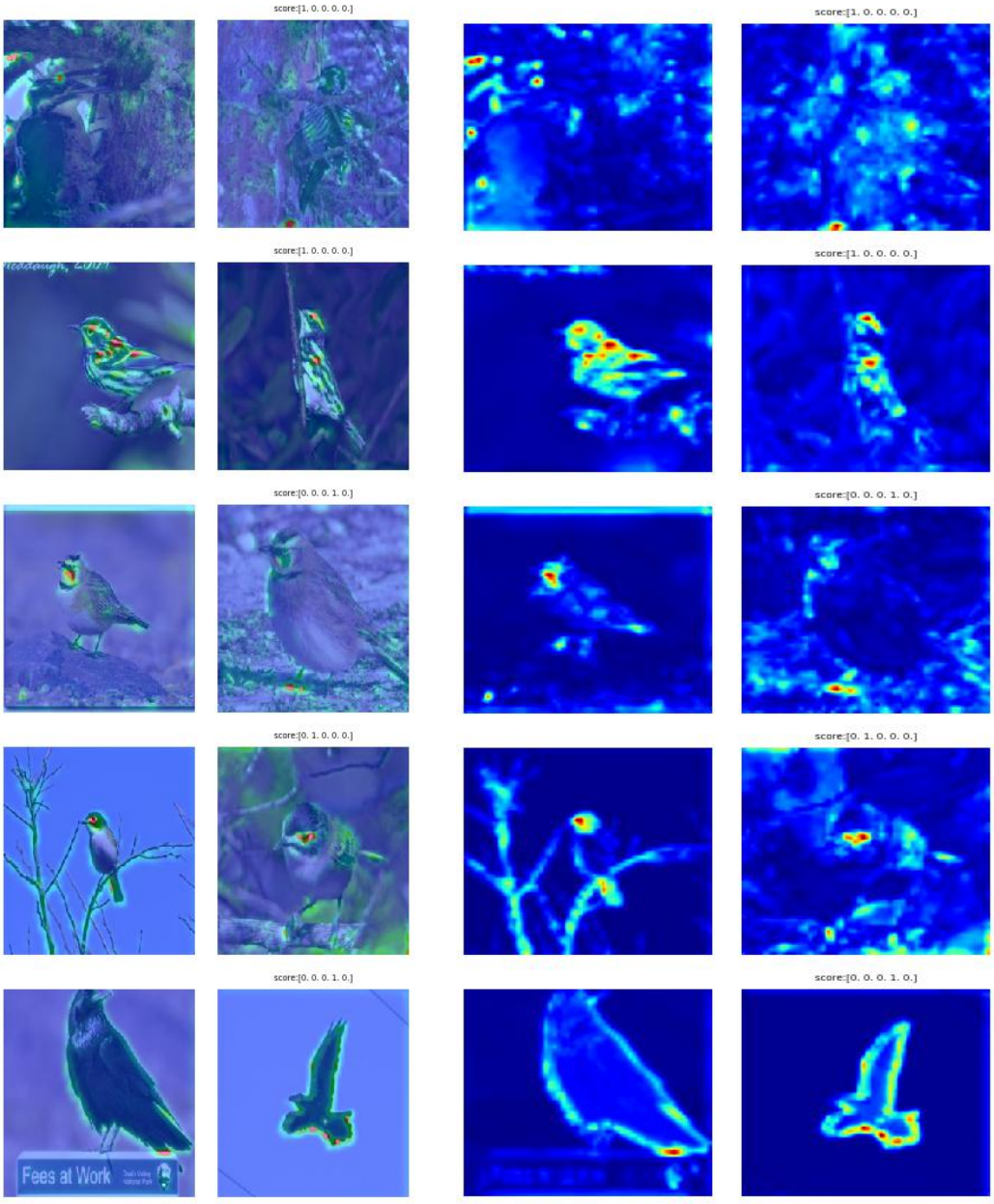
score:[1. 0. 0. 0. 0.]
score:[1. 0. 0. 0. 0.]
score:[1. 0. 0. 0. 0.]
score:[1. 0. 0. 0. 0.]
score:[0. 0. 0. 1. 0.]
score:[0. 0. 0. 1. 0.]
score:[0. 1. 0. 0. 0.]
score:[0. 1. 0. 0. 0.]
score:[0. 0. 0. 1. 0.]
score:[0. 0. 0. 1. 0.]
Fees at Work

图 4

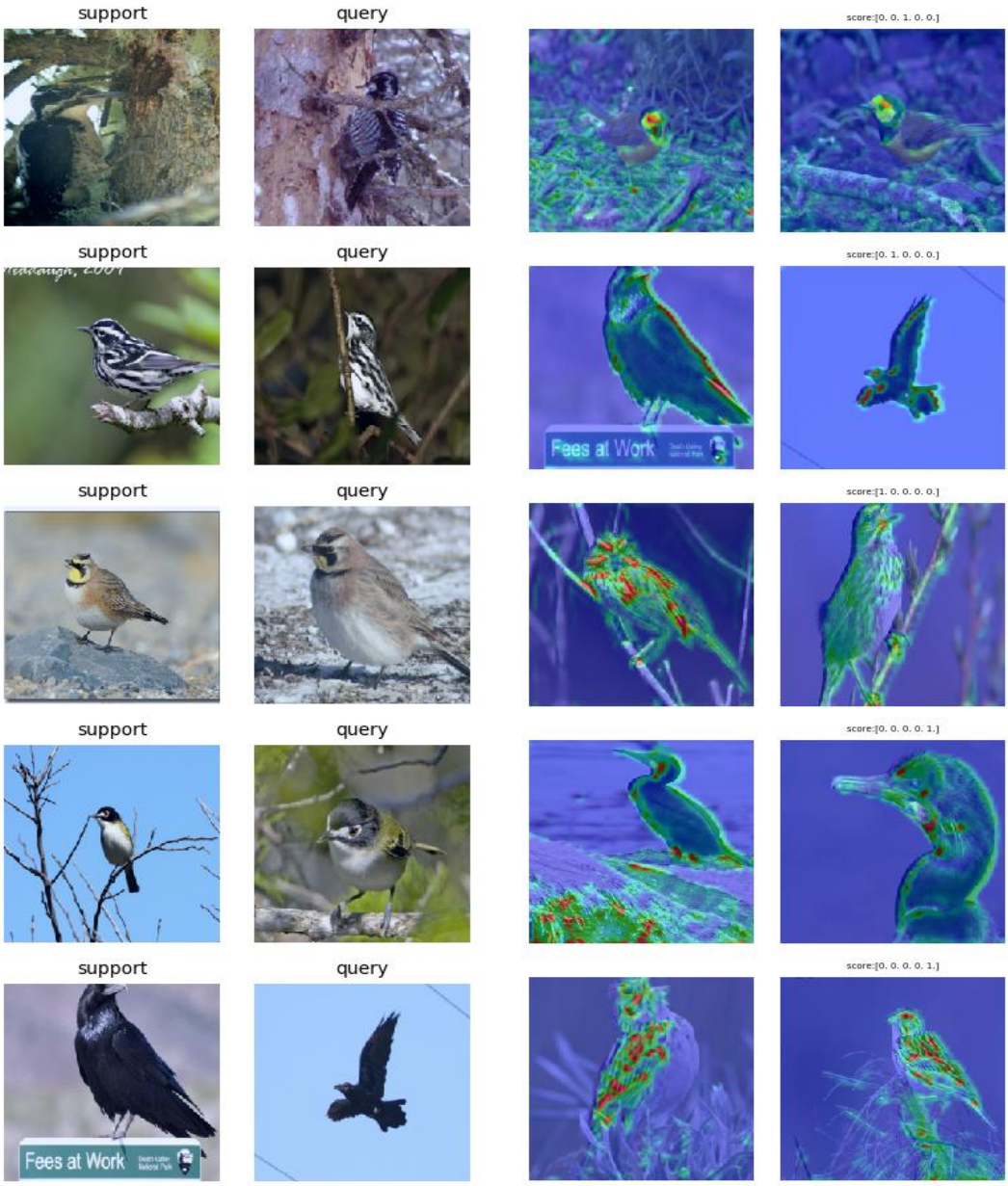
support
query
score:[0. 0. 1. 0. 0.]
support
query
score:[0. 1. 0. 0. 0.]
Fees at Work
support
query
score:[1. 0. 0. 0. 0.]
support
query
score:[0. 0. 0. 0. 1.]
support
query
score:[0. 0. 0. 0. 1.]
Fees at Work

图 5

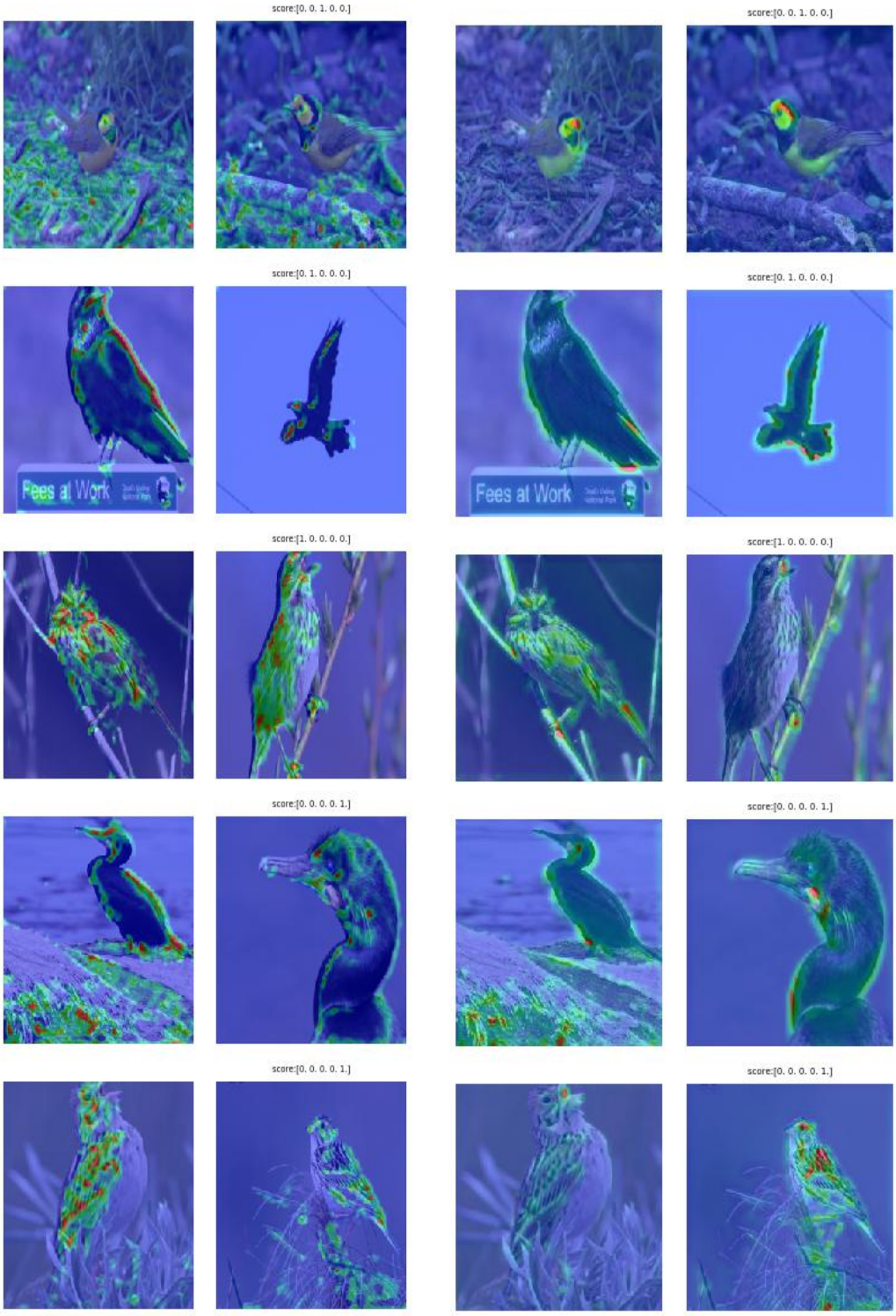
score:[0. 0. 1. 0. 0.]
score:[0. 0. 1. 0. 0.]
score:[0. 1. 0. 0. 0.]
score:[0. 1. 0. 0. 0.]
Fees at Work
Fees at Work
score:[1. 0. 0. 0. 0.]
score:[1. 0. 0. 0. 0.]
score:[0. 0. 0. 0. 1.]
score:[0. 0. 0. 0. 1.]
score:[0. 0. 0. 0. 1.]
score:[0. 0. 0. 0. 1.]

图 6

参考文献

[1] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and KlausRobert Muller. How to explain individual classification decisions. ¨ Journal of Machine Learning Research, 11(Jun):1803–1831, 2010.

[2] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. arXiv e-prints, 2014.

[3] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3319–3328. JMLR. org, 2017.

[4] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg. Smoothgrad: ´ removing noise by adding noise. arXiv:1706.03825 [cs, stat], 2017.

[5] Sebastian Bach, Alexander Binder, Gregoire Montavon, Frederick Klauschen, Klaus-Robert M ´ uller, ¨ and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE, 10(7), 2015.

[6] Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus-Robert ´ Muller. Evaluating the visualization of what a deep neural network has learned. ¨ IEEE transactions on neural networks and learning systems, 28(11):2660–2673, 2016.

[7] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3145–3153. JMLR.org, 2017.

[8] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision, pp. 818–833. Springer, 2014.

[9] Schulz K, Sixt L, Tombari F, et al. Restricting the flow: Information bottlenecks for attribution[J]. arXiv preprint arXiv:2001.00396, 2020.

[10] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural computation, 7(6):1129–1159, 1995.

[11] Teuvo Kohonen. The self-organizing map. Neurocomputing, 21(1-3):1–6, 1998.

[12] Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062, ICML'2018, 2018.

[13] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[14] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

[15] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[16] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.

[17] Hjelm R D, Fedorov A, Lavoie-Marchildon S, et al. Learning deep representations by

mutual information estimation and maximization[J]. arXiv preprint arXiv:1808.06670, 2018.
[18] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[C]//Advances in neural information processing systems. 2017: 4077-4087.
[19] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[20] Tishby, Naftali, and Noga Zaslavsky. "Deep learning and the information bottleneck principle." 2015 IEEE Information Theory Workshop (ITW). IEEE, 2015.