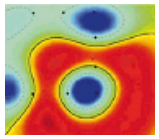
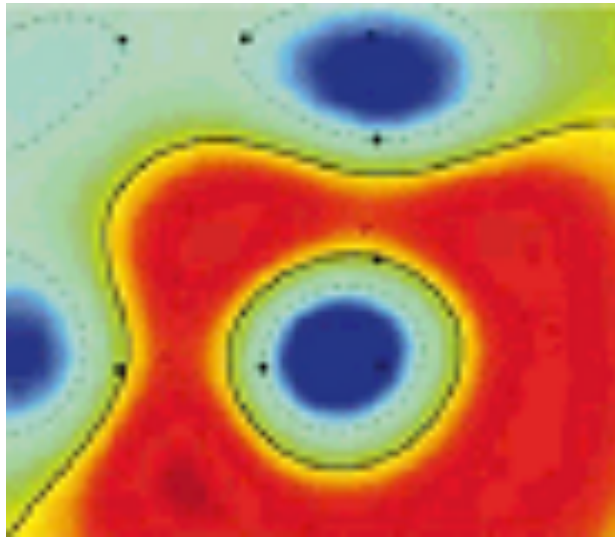


WiSe 2024/25

Deep Learning 1

Technische Universität Berlin



Lecture 6

Overfitting & Robustness (2)

Outline

Worst-Case Analysis

- The problem of adversarial examples
- Adversarial robustness

Adding Predictive Uncertainty

- Why predictive uncertainty?
- Density networks
- Ensemble models

Adding Prior Knowledge

- Translation invariance, local smoothness, etc.
- Feature reuse (transfer / multitask / self-supervised learning)

Part 1

Worst-Case Analysis

Motivations

风险规避（Risk aversion）

► 一个重大错误往往比许多小错误更具危害性。例如，由神经网络控制的系统可能对小错误具有一定的容忍度（因为它们可以随后被纠正），但无法从一个不可挽回的重大错误中恢复。

（图片：一台倒在水池中的机器人，周围有几名工作人员在查看）

Risk aversion

- One big error can often be more harmful than many small errors, e.g. a system being controlled by a neural network may be tolerant to small errors (which can be corrected subsequently), but not to a big error from which one cannot recover.

Adversarial components

- Even though the neural network may perform well on average, an adversary may craft inputs that steer the ML system towards the worst-case decision behavior.

对抗性组件（Adversarial components）

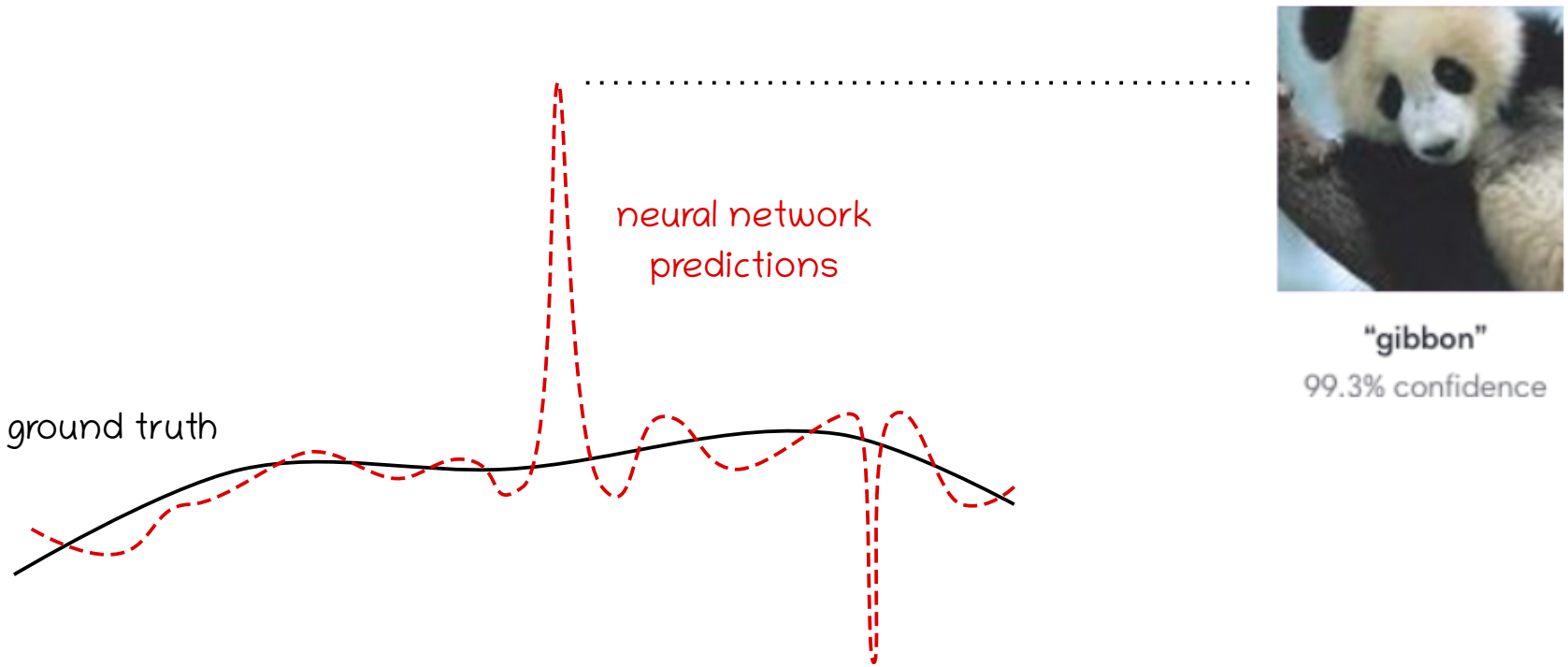
► 即使神经网络整体上表现良好，攻击者仍然可以精心设计输入，使机器学习系统在最坏情况下做出错误决策。

Worst-Case Analysis

最坏情况分析（Worst-Case Analysis）

（图表：黑色的"真实值"曲线平缓，而红色的"神经网络预测"曲线出现剧烈波动，显示网络在某些输入下的异常响应）

（图片：一只熊猫被神经网络错误识别为"长臂猿"，并给出了99.3%的置信度）



Typical causes of large errors:

- High dimensionality of input space allows to finely craft patterns to which the network responds highly.
- High depth/nonlinearity implies that the function is locally steeper than necessary.

导致大误差的典型原因：

- 输入空间的高维性允许攻击者精心设计特定模式，使得神经网络对这些模式异常敏感。
- 网络的高深度/非线性特性意味着其函数局部变化剧烈，可能比实际需要的更陡峭。

Example: Adversarial Examples

- Carefully crafted nearly invisible perturbations of an existing data point can cause the prediction of a neural network to change drastically, while leaving almost no trace of the attack.

示例：对抗样本（Example: Adversarial Examples）

► 精心设计的、几乎不可见的扰动可以导致神经网络对原始数据点的预测发生剧烈变化，同时几乎不会留下任何攻击的痕迹。

（图片：展示了一系列对抗攻击示例。左侧是原始图像，右侧是加入微小扰动后的图像，这些变化几乎无法察觉，但神经网络的预测结果发生了严重偏差。例如，一张正确识别为“建筑物”的图像，在添加微小扰动后被误认为是“鸵鸟”。）

► 这一问题在多个应用场景中是一个严重的隐患，例如生物识别、交通标志识别等。

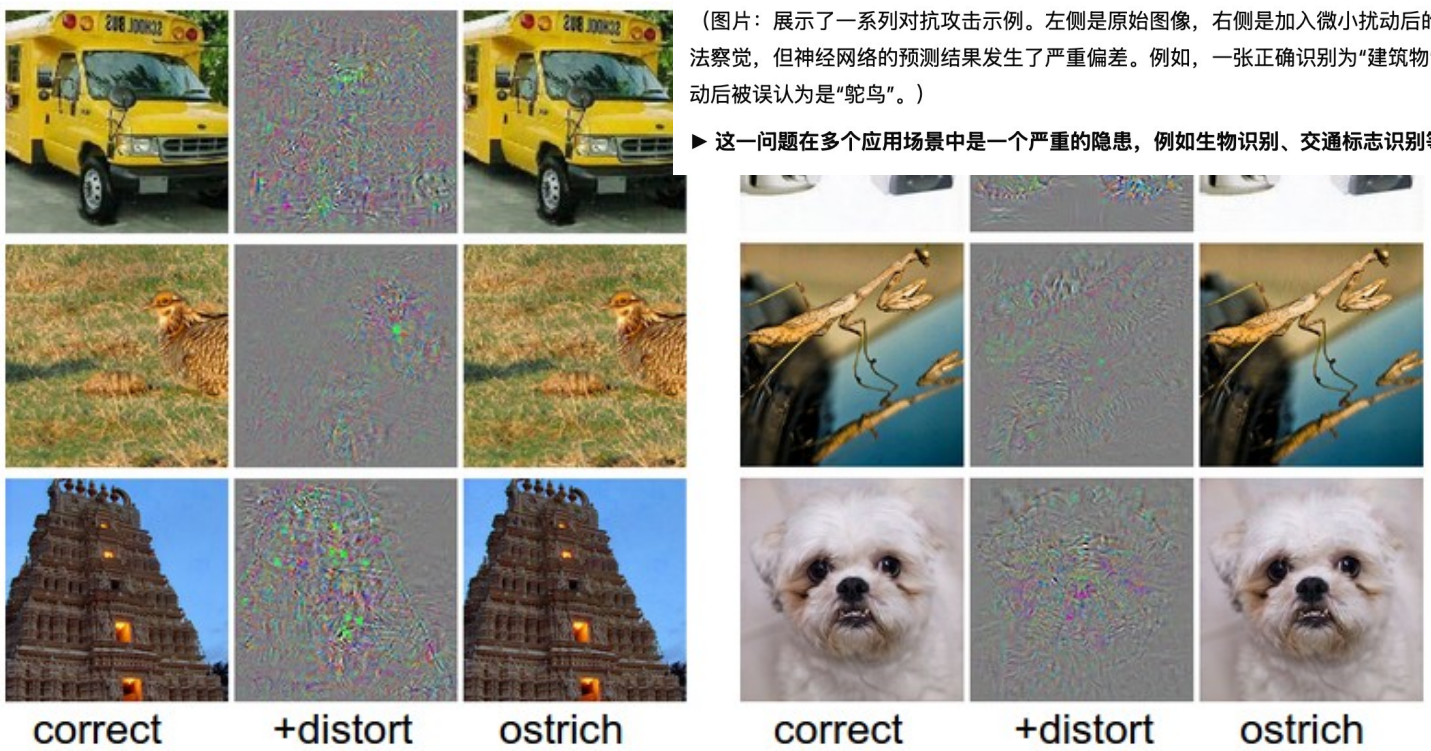


Image source: https://arxiv.org/abs/1312.6199

- Serious concern in various applications (e.g. biometric identification, reading traffic signs).

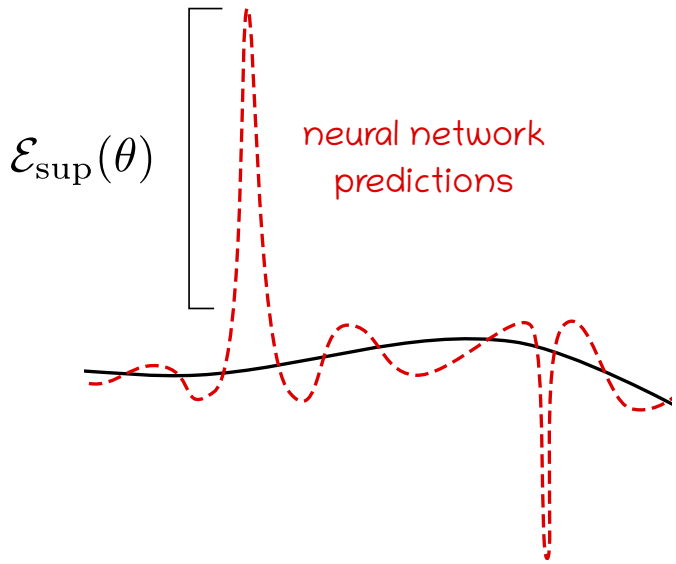
Addressing Worst-Case Behavior

Enhanced Regularization:

- Search for high local variations of the decision function and add these variations as a term of the error function to minimize.
- In practice, this can take the form of generating adversarial examples, and forcing them to be predicted in the same way as the original data.
- More generic approaches based on Lipschitz-continuity (e.g. spectral norm regularization) can also be used.

Data Preprocessing:

- In practice, one can also address worst-case behavior by applying dimensionality reduction (e.g. blurring images) before applying the neural network.



应对最坏情况行为（Addressing Worst-Case Behavior）

增强正则化（Enhanced Regularization）：

► 搜索决策函数的局部高变化区域，并将这些变化作为误差函数的一部分，以最小化误差。

► 在实践中，这可以通过生成对抗样本，并强制神经网络对其做出与原始数据一致的预测来实现。

► 还可以采用更通用的方法，例如基于Lipschitz连续性的正则化（如谱范数正则化）来减少模型的不稳定性。

数据预处理（Data Preprocessing）：

► 在实践中，也可以通过应用降维技术（例如模糊化图像）来减少最坏情况行为，然后再将数据输入神经网络。

（图表：黑色"真实值"曲线平稳，而红色"神经网络预测"曲线出现剧烈波动，表明网络在某些输入上的极端不稳定性。）

Part 2

Predictive Uncertainty

Predictive Uncertainty

Practical motivations:

- Understand when we can trust the model in a more precise way than just looking at the overall error.
- Enables the user to be prompted when the model is unsure, in which case, the user can decide e.g. to collect more data, or to perform the prediction manually.

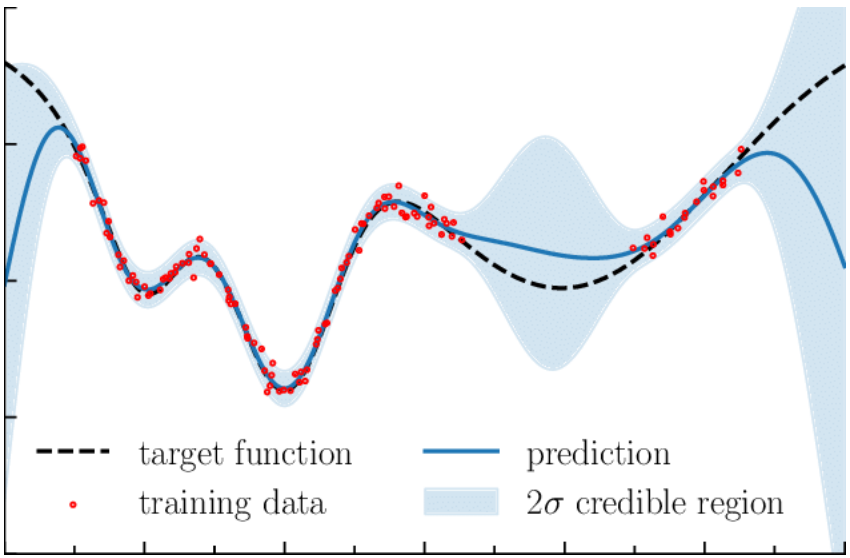


Image source: https://doi.org/10.1103/PhysRevD.98.063511

预测不确定性（Predictive Uncertainty）

实际动机（Practical motivations）：

▶ 了解何时可以信任模型，并比仅观察整体误差更精确地评估模型的可靠性。

▶ 当模型不确定时，可以提醒用户，从而用户可以决定是否收集更多数据，或者手动执行预测。

（图表：黑色虚线代表目标函数，红点代表训练数据，蓝色实线代表预测值，浅蓝色区域表示 ±2σ 可信区域。）

Predictive Uncertainty

预测不确定性（Predictive Uncertainty）

方法 1（Approach 1）：

▶ 在神经网络中显式编码不确定性估计，即使用一个输出神经元预测感兴趣的数值，并使用第二个输出神经元预测与此预测相关的不确定性。

▶ 例如，输出可以被建模为服从正态分布 $y \sim \mathcal{N}(\mu, \sigma^2)$，其中 μ 和 σ 是神经网络的两个输出变量。（关于如何训练这些模型，将在第 7 讲中介绍。）

Approach 1:

- Explicitly encoding the uncertainty estimate in the neural network, i.e. have one output neuron for predicting the actual value of interest, and a second output neuron for predicting the uncertainty associated to this prediction.
- For example, one predicts that the output is distributed according to the random variable $y \sim \mathcal{N}(\mu, \sigma^2)$ where μ and σ are the two neural network outputs. (How to train these models will be presented in Lecture 7.)

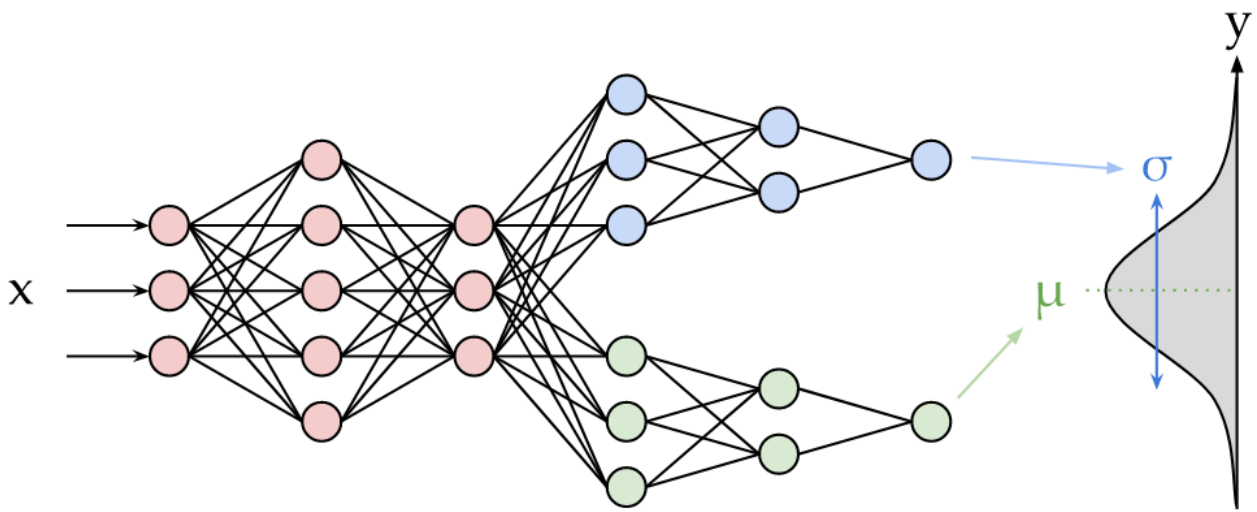


Image source: https://brendanhasz.github.io/2019/07/23/bayesian-density-net.html

Predictive Uncertainty

▶ 训练多个神经网络的集成（ensemble），并通过比较每个神经网络的预测结果来衡量不确定性。例如，对于一个包含 n 个神经网络的集成，其中每个网络的输出分别为 $o_1, o_2, ..., o_n$，可以计算以下两个统计量：

Approach 2:

- Train an ensemble of neural networks and measure prediction uncertainty as the discrepancy of predictions of each network in the ensemble, e.g. for an ensemble of n neural networks with respective outputs $o_1, \ldots, o_n$, we generate the two aggregated outputs

$$\mu = \mathrm{avg}(o_1, \ldots, o_n) \quad \text{and} \quad \sigma = \mathrm{std}(o_1, \ldots, o_n).$$

that represent the final prediction and its uncertainty.

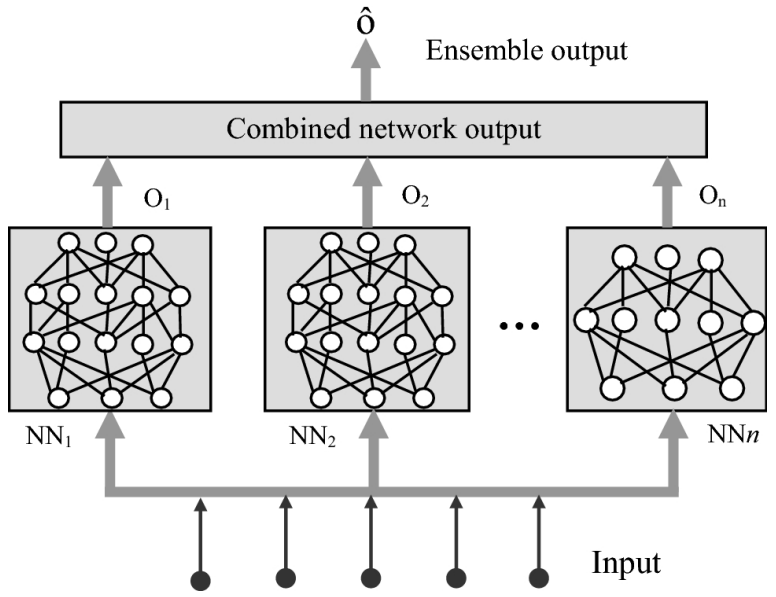


Image source: https://doi.org/10.1007/s00521-019-04359-7

Predictive Uncertainty

Approach 2 (cont.):

▶ 集成中的每个神经网络可能具有不同的初始化方式，可能接收不同的输入特征，并可能在不同的数据子集上进行训练。

▶ 因此，不确定性可以被理解为神经网络的随机初始化、特征选择和数据样本的影响。

▶ 集成越异质化（即内部差异越大），其对不确定性的估计值就越高。

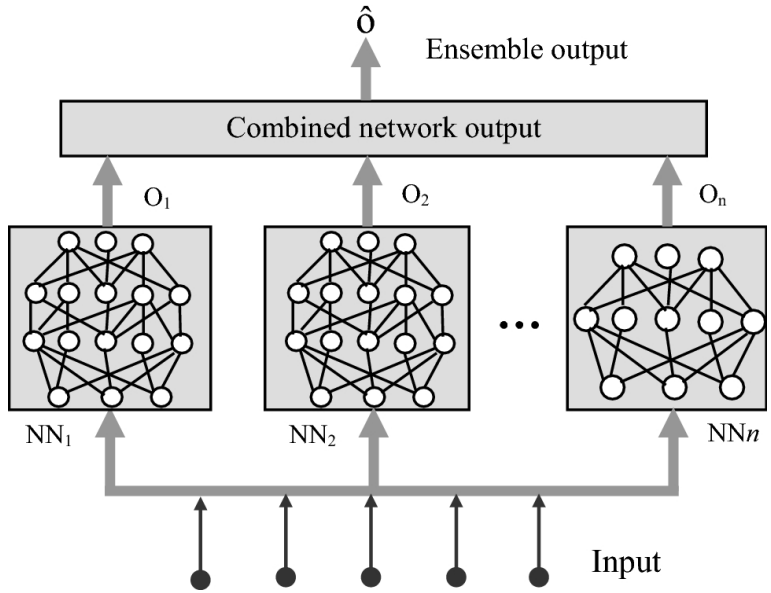


Image source: https://doi.org/10.1007/s00521-019-04359-7

- Each network in the ensemble may have a different initialization, may receive different input features, and may be trained on different subsets of data.
- Uncertainty can then be understood as the effect of neural network random initialization, feature selection, data sample.
- The more heterogeneous the ensemble, the higher the estimate of uncertainty.

Part 3

Beyond Regularization: Prior Knowledge

Prior Knowledge

Recap:

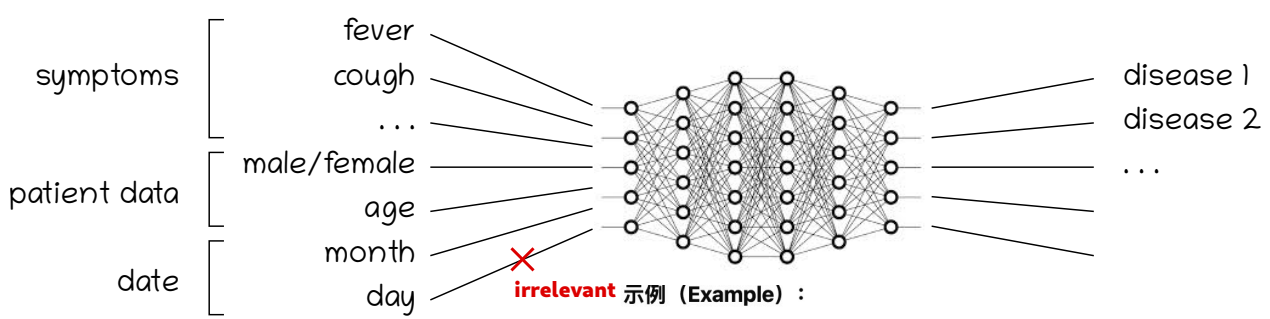
- Machine learning is affected by data quality issues (e.g. scarcity of data or labels, spurious correlations in the dataset, under-representation of some part of the distribution, shift between data available for training and data when deployed).

Idea:

- **There is no point to learn from the data what we already know.** What we already know (our prior knowledge) should ideally be hard-coded into the model.

Example:

- In specific tasks, certain features are known to have no effect on the quantity to predict. It is better to not give them as input to the neural network.



先验知识（Prior Knowledge）

回顾（Recap）：

► 机器学习受到数据质量问题的影响（例如数据或标签稀缺、数据集中存在虚假相关性、某些部分的数据分布不足，以及训练数据与部署数据之间的分布偏移）。

理念（Idea）：

► **从数据中学习我们已经知道的内容是没有意义的。**
我们已经掌握的知识（即先验知识）理想情况下应该直接编码到模型中。

示例（Example）：

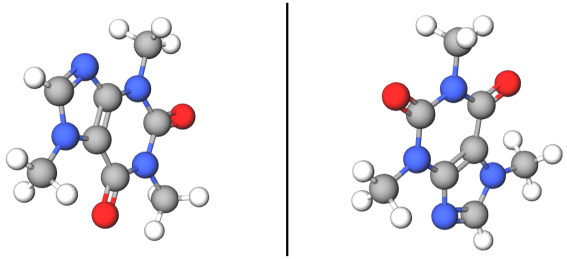
► 在某些任务中，已知某些特征对要预测的数量没有影响。因此，最好不要将这些特征作为输入提供给神经网络。

（图片示例：患者数据包含症状（如发烧、咳嗽）和其他信息（如性别、年龄、日期等），但其中某些信息（如"日期"）被标记为"无关"，不应输入神经网络。）

Physical Invariances

Example: Rotation/Translation Invariance

- Rotating or translating a molecule maintains its atomization energy constant.



- Rotation invariance can be ensured e.g. by encoding the molecule by the pairwise distances between its atoms rather than its 3d coordinates, and feeding these distances to a neural network.

$$\Phi(\mathbf{r}) = \begin{pmatrix} \|\mathbf{r}_1 - \mathbf{r}_2\|, & \|\mathbf{r}_1 - \mathbf{r}_3\|, & \dots & \|\mathbf{r}_1 - \mathbf{r}_L\| \\ & \|\mathbf{r}_2 - \mathbf{r}_3\|, & & \|\mathbf{r}_2 - \mathbf{r}_L\| \\ & & \ddots & \vdots \\ & & & \|\mathbf{r}_{L-1} - \mathbf{r}_L\| \end{pmatrix}$$

物理不变性（Physical Invariances）

示例：旋转/平移不变性（Example: Rotation/Translation Invariance）

► 旋转或平移分子不会改变其原子化能量。

► 旋转不变性可以通过将分子表示为原子之间的两两距离，而不是其 3D 坐标，并将这些距离输入到神经网络中来实现。

（图片示例：两个不同角度的相同分子结构，展示了旋转不影响其特性。数学公式展示了距离矩阵的计算方式。）

Physical Invariances

Example: Modeling interaction between two molecules

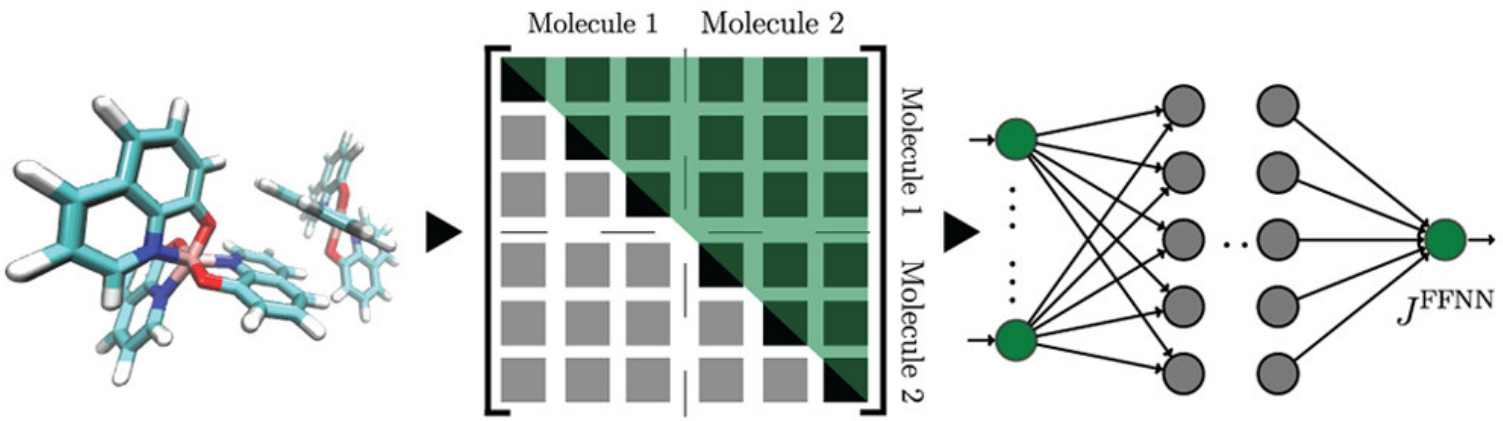


Image source: https://doi.org/10.1021/acs.jctc.8b01285

- Distances are fed as input to a plain neural network.
- Work as long as atom of the molecules can be indexed (i.e. approach stops working when the molecules received as input are of arbitrary shape and size).

物理不变性（Physical Invariances）

示例：建模两个分子之间的相互作用（Example: Modeling Interaction Between Two Molecules）

（图片示例：分子相互作用示意图，展示了如何构造输入矩阵并输入神经网络。）

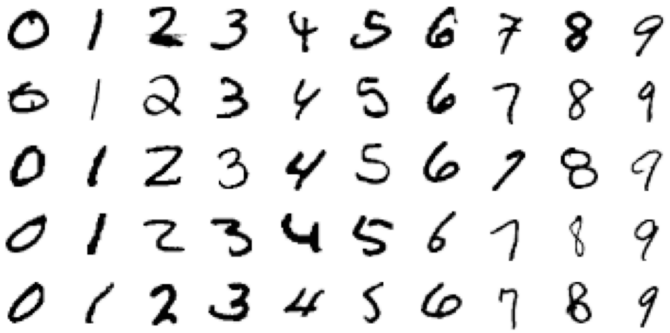
▶ 计算分子之间的距离，并将其作为输入提供给普通的神经网络。

▶ 该方法适用于分子中的原子可以被索引的情况（即，该方法在输入的分子具有任意形状和大小时可能会失效）。

Soft Invariances

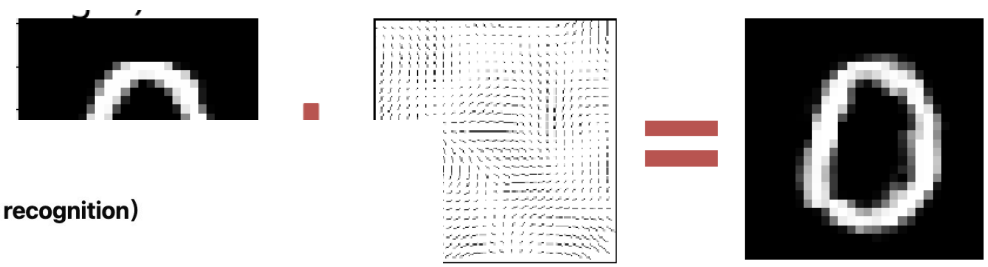
Example: Handwritten digits recognition

- Rotating digits by a few degrees usually does not change class membership.
- Some exceptions, e.g. rotating a '1' may transform it into a '7'.



Approaches to build invariance:

- Use purposely designed neural network architectures. E.g. scattering networks, pooling networks, etc.
- Augment the dataset with elastic distortions, and train the neural network on the extended dataset.



A practical guide to handwritten digits classifier & dataset

-github-1.html

软不变性（Soft Invariances）

示例：手写数字识别（Example: Handwritten digits recognition）

► 轻微旋转数字通常不会改变其类别。

► 但也存在例外情况，例如，将"1"旋转可能会使其变成"7"。

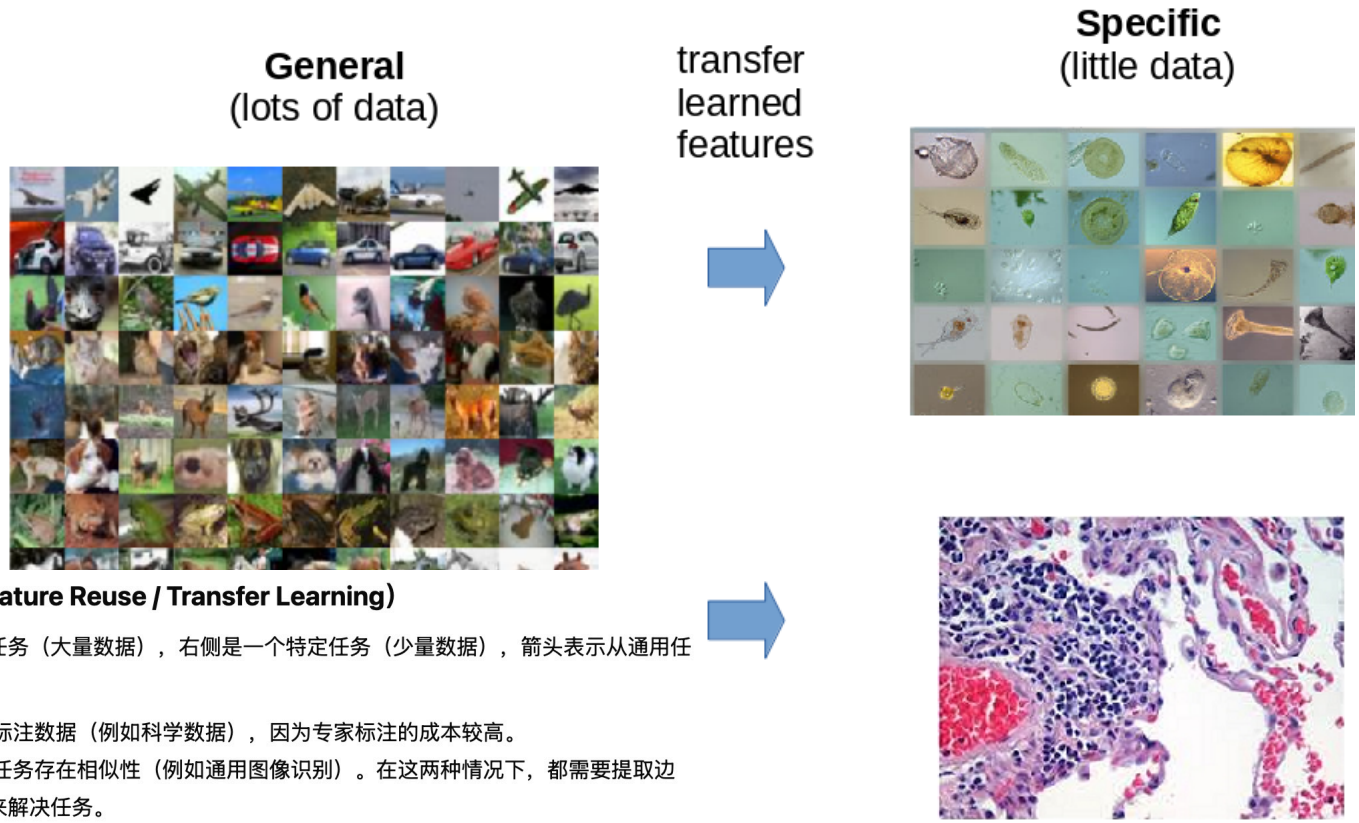
构建不变性的方式（Approaches to build invariance）：

► 采用专门设计的神经网络架构，例如散射网络（scattering networks）、池化网络（pooling networks）等。

► 通过弹性变形（elastic distortions）扩充数据集，并在扩展后的数据集上训练神经网络。

（图片示例：展示了数字旋转对分类的影响，以及如何通过弹性变形增强数据集。）

Feature Reuse / Transfer Learning



特征重用 / 迁移学习（Feature Reuse / Transfer Learning）

（图片示例：左侧是一个通用任务（大量数据），右侧是一个特定任务（少量数据），箭头表示从通用任务迁移特征到特定任务。）

▶ 某些任务本质上具有较少的标注数据（例如科学数据），因为专家标注的成本较高。

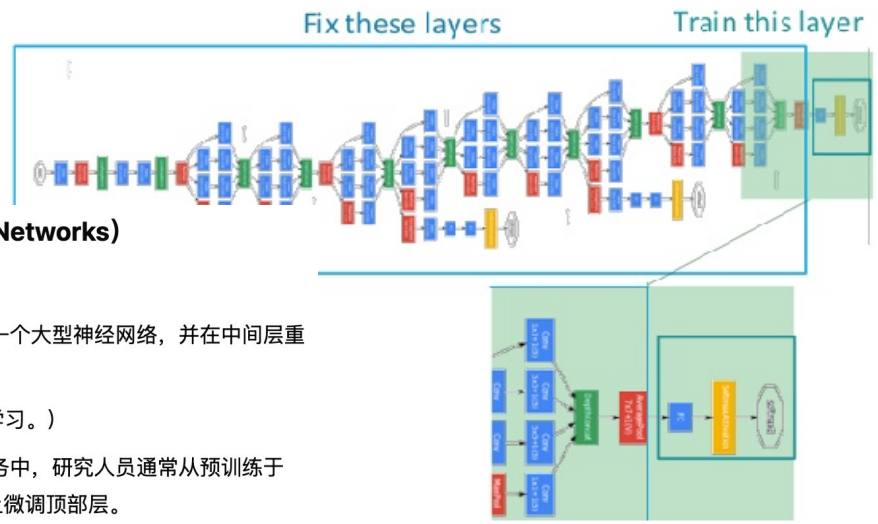
▶ 但它们与数据量更大的其他任务存在相似性（例如通用图像识别）。在这两种情况下，都需要提取边缘检测器或颜色检测器等特征来解决任务。

- Certain tasks have intrinsically little annotated data (e.g. scientific data), due to the cost of labeling by an expert.
- However, they have similarities with other tasks with much more data (e.g. general-purpose image recognition). In both cases, one needs to extract features such as edge or color detectors to solve the task.

Transfer Learning with Deep Networks

Approach:

- When two tasks are *related*, we can train a big neural network on the first task with abundant data, and reuse features in intermediate layer for the task of interest.



深度网络中的迁移学习（Transfer Learning with Deep Networks）

方法（Approach）：

► 当两个任务**相关**时，我们可以在拥有大量数据的第一个任务上训练一个大型神经网络，并在中间层重用特征以应用到目标任务中。

（图片示例：展示了如何固定部分层，并在高层重新训练以进行迁移学习。）

► 这种类型的迁移学习在应用研究中非常常见。例如，在图像识别任务中，研究人员通常从预训练于 ImageNet 的先进视觉神经网络（如 ResNet）开始，然后在特定任务上微调顶部层。

Image Source: Mark Chang, Applied Deep Learning 11/03 Convolutional Neural Networks

- This type of transfer learning is very common in applied research. For image recognition tasks, researchers typically start from a state-of-the-art network for vision (e.g. ResNet) pretrained on ImageNet, and retrain the top layers on the specific task.

How to Generate Useful Features

Classical approach:

- Learn a model to classify a more general task (e.g. image recognition).

Self-supervised learning approach

- Create an artificial task where labels can be produced automatically, and whose solution requires features that are needed for the task of interest (more in DL2).

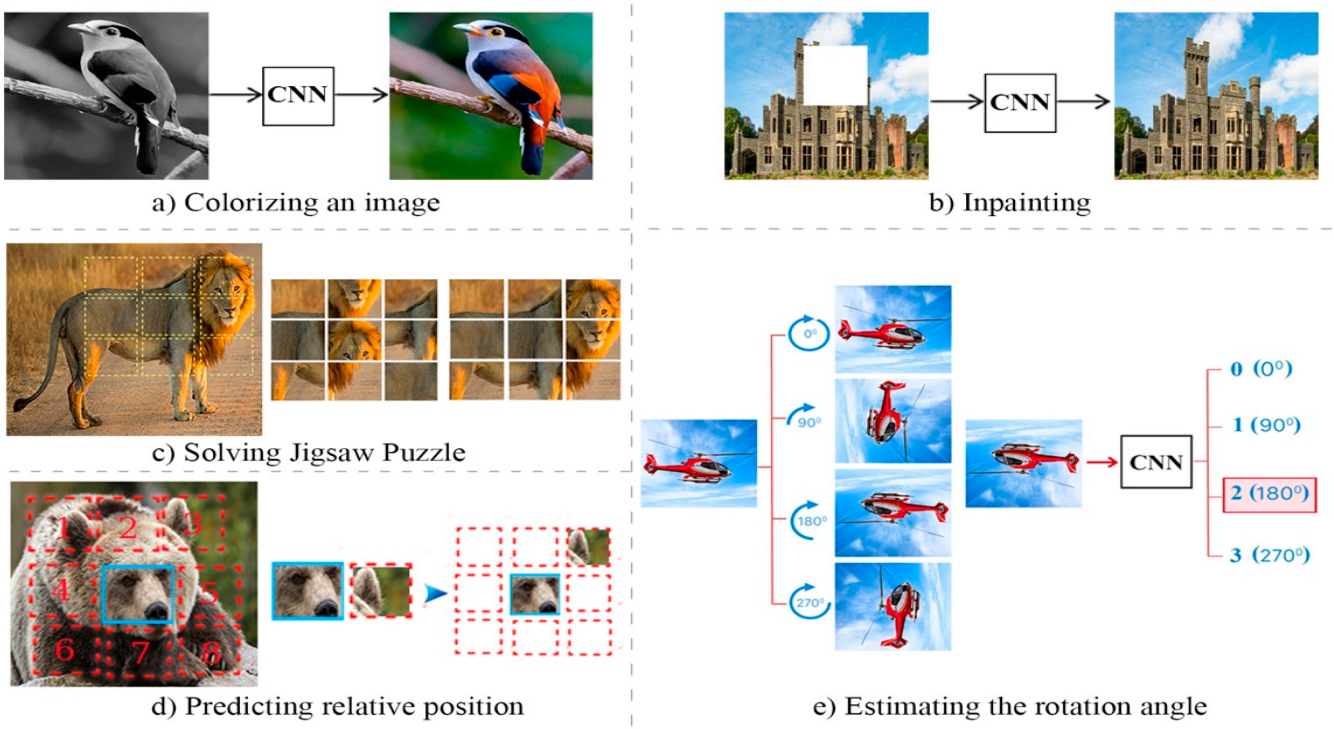


Image source: https://doi.org/10.3390/e24040551

Summary

Summary

▶ 在实践中，期望的预测准确性可能不是最相关的量。系统的真正实用性通常更好地由其**最坏情况下的性能**来衡量。

▶ 通常希望在神经网络模型中引入一定程度的**预测不确定性**，以便网络能够告诉用户**何时信任预测结果，何时不信任。**

▶ **没有必要从数据中学习我们已经知道的内容。** 先验知识（如**不变性、共享特征**）可以被引入神经网络中，从而使模型**减少过拟合**，并且**更加健壮。**

- In practice, expected prediction accuracy may not be the most relevant quantity. The true practical usefulness of a system is often better characterized by its worst-case performance.
- It is often desirable to equip neural network models with some measure of predictive uncertainty so that the network can tell the user when to trust and when not to trust the prediction.
- There is no point to learn what we already know. Prior knowledge (e.g. invariances, shared features) can be introduced in neural networks. As a result, the model becomes less affected by overfitting and also more robust.