Semester Thesis: Measuring leakage in Concept-based methods

Mikael Makonnen, Moritz Vandenhirtz

November 19, 2024

1 Background & Motivation

Concept Bottleneck Models Concept bottleneck models (CBM) Koh et al. (2020); Lampert et al. (2009); Kumar et al. (2009) are a simple class of interpretable neural networks typically trained on data points $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{y})$, comprising the covariates $\boldsymbol{x} \in \mathcal{X}$ and targets $\boldsymbol{y} \in \mathcal{Y}$ additionally annotated by the concepts $\boldsymbol{c} \in \mathcal{C}$. Consider a neural network $f_{\boldsymbol{\theta}}$ parameterised by $\boldsymbol{\theta}$ and a slice $\langle g_{\boldsymbol{\psi}}, h_{\boldsymbol{\phi}} \rangle$ Leino et al. (2018) s.t.

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x})) \tag{1}$$

for all $\boldsymbol{x} \in \mathcal{X}$, where $\hat{\boldsymbol{y}} := f_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x}))$ denote the output of the network, *i.e.* the predicted targets. CBMs enforce a concept bottleneck $\hat{\boldsymbol{c}} := h_{\boldsymbol{\phi}}(\boldsymbol{x})$: the model's final output depends on the covariates $\boldsymbol{x}$ solely through the predicted concepts $\hat{\boldsymbol{c}}$. Thus, in addition to the target prediction loss applied to the final output, $h_{\boldsymbol{\phi}}(\cdot)$ is trained to predict the ground-truth concept values.

Interpretability The interpretability of CBMs is achieved by the set of high-level, human-understandable concepts. Often, these are C binary-valued attributes, i.e. $\mathcal{C} = \{0, 1\}^C$ that can be easily detected from the covariates $\boldsymbol{x}$ and are predictive of the targets $\boldsymbol{y}$. Although CBMs make no assumptions on (anti)causal relationships among $\boldsymbol{x}$, $\boldsymbol{c}$, and $\boldsymbol{y}$, they implicitly assume that concepts $\boldsymbol{c}$ are a sufficient statistic Yeh et al. (2020) for predicting $\boldsymbol{y}$ based on $\boldsymbol{x}$ Havasi et al. (2022); Marcinkevičs et al. (2024), i.e. $\boldsymbol{y} \perp\!\!\!\perp \boldsymbol{x} \,|\, \boldsymbol{c}$.

Leakage Leakage is an instance of shortcut learning (Geirhos et al., 2020). Margeloiu et al. (2021); Mahinpei et al. (2021); Havasi et al. (2022) show that leakage occurs in cases where the conditional independence assumption does not hold. The distribution of the predicted concept values encodes more information than solely the probability of concept presence. This additional information can then be exploited by the classifier $g_{\boldsymbol{\psi}}(\cdot)$. This is an issue since the predicted concept values encode information different from the human-understandable concepts, thus, prohibiting the interpretation of the predicted probability as probability of concept presence. Mahinpei et al. (2021) show that even if the predicted concepts are not soft (*i.e.* $\boldsymbol{c} \in [0, 1]$) but hard (*i.e.* $\boldsymbol{c} \in \{0, 1\}$), leakage happens, albeit weaker. Therefore, any perception of interpretability for standard CBMs is void if $\boldsymbol{y} \perp\!\!\!\perp \boldsymbol{x} \,|\, \boldsymbol{c}$ is not fulfilled, which is often the case in real-world problems. Examples of works (unintentionally) committing this fallacy are Espinosa Zarlenga et al. (2022); Marconato et al. (2022); Ismail et al. (2023). To understand how strongly the interpretability of concept probabilities is restricted, we need a metric that is able to measure the leakage within these concept embeddings.

2 Related Work

To measure leakage, Zarlenga et al. (2023) propose metrics that estimate the degree of excessive information with respect to other concepts, which they call impurity. To resolve leakage, Margeloiu et al. (2021) recommend using the *independent* training procedure with hard concepts. However, this comes at the cost of decreasing performance since the encoder and predictor head can not communicate anymore. Thus, Havasi et al. (2022) propose to include a hard side-channel, in which the additional information can be learned explicitly, as well as an autoregressive structure over the hard concept predictions, such that their correlations can be captured. At intervention time, they use importance-weighted MCMC sampling to implicitly learn the effect of a concept intervention on the other concepts. It will be interesting to see whether their approach fully eradicates leakage.

3 Methods

Consider a neural network $NN_{\boldsymbol{\theta}}$ parameterised by $\boldsymbol{\theta}$ and a slice $\langle g_{\boldsymbol{\psi}}, h_{\boldsymbol{\phi}} \rangle$ Leino et al. (2018) s.t.

$$NN_{\boldsymbol{\theta}}(\boldsymbol{x}) = g_{\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x})) \tag{2}$$

For sake of intuition, think of it as a CBM, where $\boldsymbol{z} = h_{\boldsymbol{\phi}}(\boldsymbol{x}) = \hat{\boldsymbol{c}}$ is trained via the prediction of concepts, but this formulation allows for a more general interpretation.

What we are interested in for leakage, is the information contained within $\boldsymbol{z}$, which is informative for the label $\boldsymbol{y}$ but independent/non-informative of concepts $\boldsymbol{c}$:

$$I(\boldsymbol{z}; \boldsymbol{y} \mid \boldsymbol{c}) = H(\boldsymbol{y} \mid \boldsymbol{c}) - H(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{c})$$

Estimating $H(\boldsymbol{y} \mid \boldsymbol{c})$ and $H(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{c})$ is the goal of this thesis. A straightforward approximation is

$$H(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{c}) = \mathbb{E}[-\log p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{c})] \approx -\frac{1}{N} \sum_{i=1}^{N} \log g_{a,\boldsymbol{\psi}}(h_{\boldsymbol{\phi}}(\boldsymbol{x}_i), \boldsymbol{c}_i)_{y_i}, \tag{3}$$

$$H(\boldsymbol{y} \mid \boldsymbol{c}) = \mathbb{E}[-\log p(\boldsymbol{y} \mid \boldsymbol{c})] \approx -\frac{1}{N} \sum_{i=1}^{N} \log g_{b,\boldsymbol{\psi}}(\boldsymbol{c}_i)_{y_i}, \tag{4}$$

where $g_{a,\boldsymbol{\psi}}$ and $g_{b,\boldsymbol{\psi}}$ are two classifiers trained to predict $\boldsymbol{y}$ from $\boldsymbol{z}$, $\boldsymbol{c}$ and from $\boldsymbol{c}$, respectively.