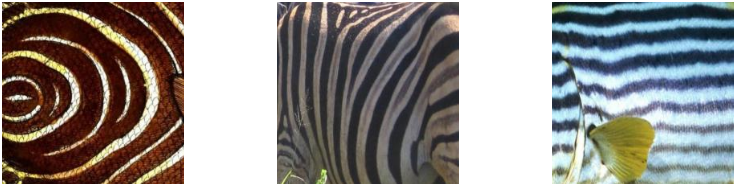
8. Concept-based Explainable AI

Introduction

- **So far:** feature-attribution methods
- $\approx$ compute some measure of importance for each feature
- not entirely satisfying, especially if many features (*e.g.*, images)
- **Another approach:** higher-level attributes used by the model ($=$ concepts)
- either directly used by the model or inferred after training
- **What is a concept?**
  - symbolic concepts;
  - unsupervised concepts basis;
  - textual concepts;
  - ...

Symbolic concepts

- **Informal definition:** high-level abstractions
- **Example:** class zebra $\rightarrow$ `striped` concept
- generally associated to human-annotated sets of examples
- $\Rightarrow$ costly + restrictive
- **Example:** image-classification
  - patches of images, someone says whether concept present or not
  - class-level annotation

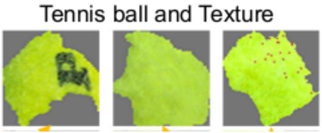


- **Figure:** images corresponding to the `striped` concept from from the `Broden`[80] dataset

[80] Bau et al., *Network Dissection: Quantifying interpretability of deep visual representations*, CVPR, 2017

Unsupervised concept basis

- **Informal definition:** cluster of similar examples or parts of examples
- **Example:** ACE[81] explanation for tennis ball



- generally extracted from some latent representation *via* clustering[82]
- **Important:** do not necessarily coincide with human-defined concepts!

[81] Ghorbani et al., *Towards Automatic Concept-based Explanations*, NeurIPS, 2019
[82] Chapter 14.3 of Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning*, Springer, 2004

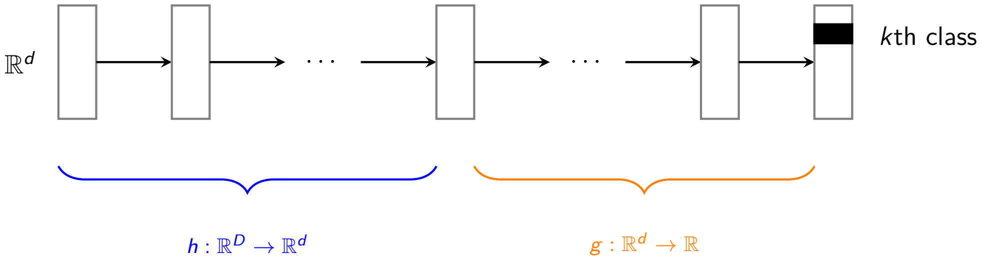
A typology of concept-based XAI

- **Main categories:**[83]
  - **Class-concept relations:** quantifying relationship between pre-determined concept and output class of a model
  - **Node-concept association:** quantifying relationship between pre-determined concept and inner node of a model
  - **Concept-visualization:** visualization in terms of input features

[83] Poeta, Ciravegna, et al., *Concept-based Explainable Artificial Intelligence: A Survey*, preprint, 2023

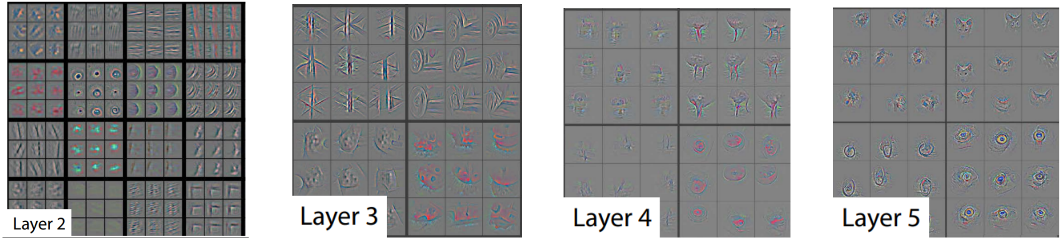
More on latent representation

- **Key ingredient in the concept-based literature:** intermediate representation of the input by the network
- **Notation:** $f : \mathbb{R}^D \to \mathbb{R}$ corresponds to logit of class k of our model
- set $f = g \circ h$, with $h : \mathbb{R}^D \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}$
- **Schematically:**



Which layer to choose?

- **Intuition:** first layers = low-level visual features
- the deeper we go, the higher the chances of finding high-level concepts are
- **Typical choice:** last convolutional layer



- **Figure:** visualizing top activations of a simili AlexNet from random samples[84]

[84] Zeiler and Fergus, *Visualizing and Understanding Convolutional Networks*, ECCV, 2014

8.1. Concept Activation Vectors

Concept Activation Vectors

- let us look at a second method: TCAV[85]
- **Big picture,** for a given example ξ:
  1. get concept + random examples;
  2. compute their latent representation;
  3. train a **linear classifier** in the layer with normal vector (V_C);
  4. compute $\nabla_{h(\xi)} g$;
  5. compute $S := \langle \nabla_{h(\xi)} g, V_C \rangle$.
- **Linear classifier** = logistic regression

[85] Kim et al., *Interpretability beyond feature attribution: quantitative testing with concept activation vectors*, ICML, 2018

Reminder: logistic regression

- classification with labels $\mathcal{Y} = \{0, 1\}$
- however, we predict the probability of belonging to class 1
- hypothesis class:

$$\mathcal{H} = \{x \mapsto \phi(\langle w, x\rangle), w \in \mathbb{R}^d\},$$

  with ϕ the *logistic function* (aka *sigmoid* function)

$$\phi(z) = \frac{1}{1 + \mathrm{e}^{-z}}.$$

- **Intuition:** squeeze the score between 0 and 1 to transform it into a probability
- $\mathbb{P}(y = 1 \,|\, x) = \phi(w^\top x)$ and $\mathbb{P}(y = 0 \,|\, x) = 1 - \phi(w^\top x)$

Logistic function

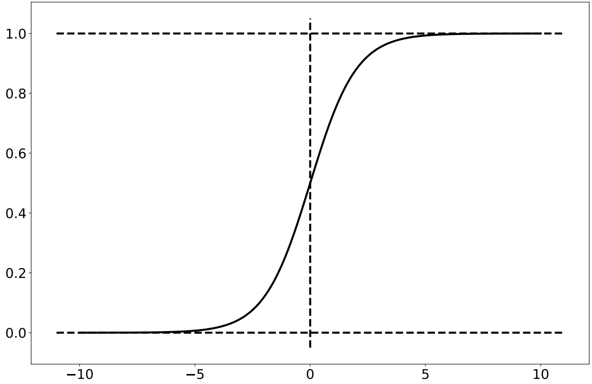
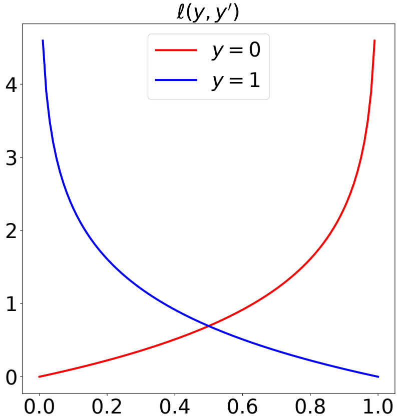


Figure: the logistic function $\phi : t \mapsto 1/(1 + \mathrm{e}^{-t})$.

Logistic loss

- **Loss function:** logistic loss (also called binary cross entropy)
- formally, for any y, y',

$$\ell(y, y') = -(1 - y)\log(1 - y') - y \log y' \, .$$



Logistic regression

- finally, logistic regression $=$ empirical risk minimization with the logistic loss
- that is, minimize for $w \in \mathbb{R}^d$

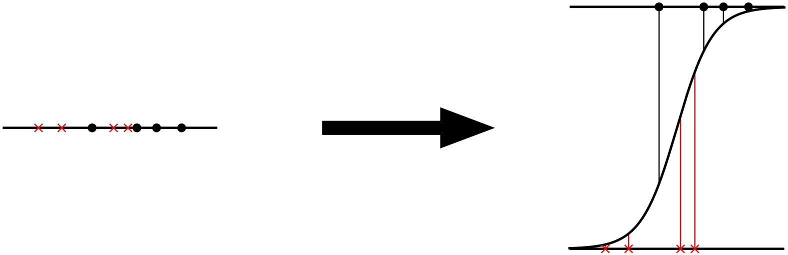
$$\hat{\mathcal{R}}(w) = \sum_{i=1}^{n} \left\{ -(1-y_i)\log(1-\phi(w^\top x_i)) - y_i \log \phi(w^\top x_i) \right\} \,.$$

- **Remark (i):** equivalent to maximum likelihood for a certain prior distribution
- **Remark (ii):** not so easy to optimize, at least simple expression for the gradient:

$$\forall j \in [d], \qquad \frac{\partial \hat{\mathcal{R}}(w)}{\partial w_j} = - \sum_{i=1}^{n} \left(y_i - \phi(w^\top x_i) \right) x_{i,j} \,.$$

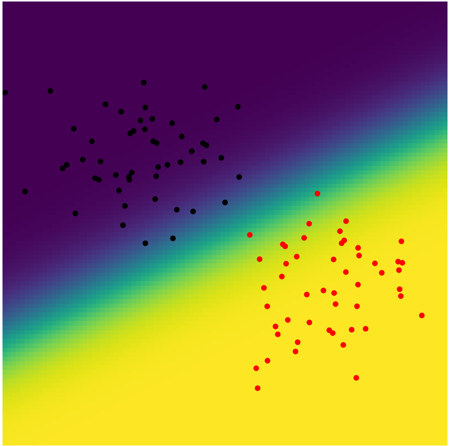
Logistic regression in dimension 1

- **Example:** in dimension one:



Logistic regression in dimension 2

- **Example:** in dimension two:



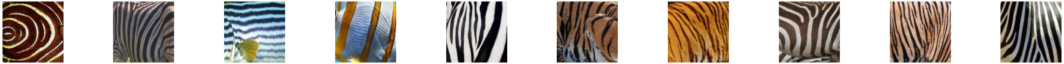
Recap

- **What happens when we call** `sklearn.linear_model.LogisticRegression`?
- `penalty` is ℓ_2 $\rightarrow$ there is regularization by default! (not much though, $C = 1$)
- `fit_intercept` is True
- `solver` is liblinear which uses *coordinate descent*
- or lbfgs (limited memory Broyden-Fletcher-Goldfarb-Shanno[86], 1989)
- not that important: variant of **gradient descent**

[86] Liu, Nocedal, *On the limited memory method for large scale optimization*, Mathematical Programming B

TCAV step 1: examples

- a **concept** is encoded as a set of n images $c_1, \ldots, c_n$:



- these images will be confronted to m images $X_1, \ldots, X_m$ chosen randomly in the train

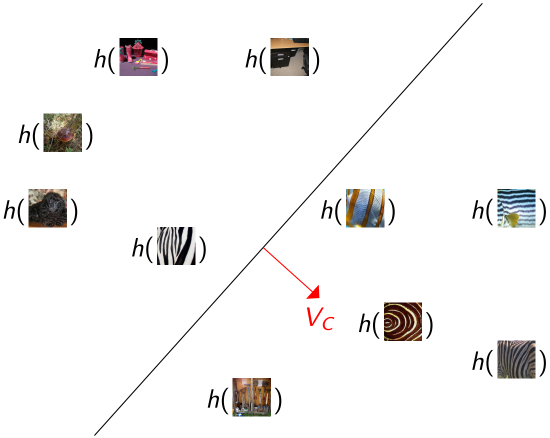
- **Remark:** typical values are $n = m = 20$

CAV step 2: latent representation

- **Reminder:** we decompose $f = g \circ h$
- we compute $h(c_i)$ and $h(X_m)$ for all $i \in [n]$ and $j \in [m]$

CAV step 3: linear classifier

- train a linear classifier (concept = positive class)
- V_C = normal vector to the separating hyperplane



CAV step 4: gradient computation

- now we consider a particular example for which we want to measure concept activation:

$$\xi =$$ 

- we compute the **gradient of the output with respect to the latent representation:**

$$\nabla_{h(\xi)} g = \left(\left. \frac{\partial g(y)}{\partial y_j} \right|_{y=h(\xi)} \right)_{j \in [d]} \in \mathbb{R}^d \, .$$

- **Intuition:** measures influence of each latent feature on the prediction

CAV step 5: compute the score

- **Definition:**

$$S_C(\xi) := \langle \nabla_{h(\xi)} g, V_C \rangle \,.$$

- **Intuition:** S_C encodes how much the concept is *activated* by the example in the considered layer
- **Examples:**

$$\xi =$$  $$\Rightarrow \qquad S_C(\xi) = 0.98\,.$$

$$\xi =$$  $$\Rightarrow \qquad S_C(\xi) = -0.07\,.$$

Testing with CAVs

- let k be a class label and $\mathcal{X}_k$ the set of inputs with that label
- we can compute scores across entire classes of inputs:

$$\mathrm{TCAV}_k := \frac{|x \in \mathcal{X}_k : S(x) > 0|}{|\mathcal{X}_k|} \in [0,1]\,.$$

- **Intuition:** fraction of k-class inputs whose activation vector is positively influenced by concept C
- **Remark:** dependency on the random examples
- Kim et al. suggest to run the experiment 500 times
- then perform two-sided t-test, with null hypothesis $= \{\mathrm{TCAV} = 0.5\}$

Reminder: statistical testing

- **Informal definition:** decide whether the observations agree with our model
- initial research hypothesis: nothing interesting happens, *e.g.*, TCAV $= 0.5$
- **Other example:** efficiency of a drug, initial hypothesis $=$ no effect
- formally, we work in a statistical model

$$\mathcal{P} = \left\{ P_\theta \text{ s.t. } \theta \in \Theta \right\} ,$$

  and **split** Θ in two *disjoint* subsets Θ_0 and Θ_1
- **Remark:** we do not require $\Theta_0 \cup \Theta_1 = \Theta$
- we define
  - $H_0 : \theta \in \Theta_0$ the null hypothesis
  - and $H_1 : \theta \in \Theta_1$ the alternative hypothesis
- given realization of $X \sim P_\theta$, **we want to decide whether H_0 or H_1 holds**

Reminder: statistical testing

Definition: we call *test* of H_0 versus H_1 any function ϕ with values in $\{0,1\}$, where ϕ is X-measurable and can depend on Θ_0 and Θ_1. When $\phi(X)=0$, we conserve H_0, when $\phi(X)=1$ we *reject* H_0.

- **Remark:** any test can be written $\phi(X) = \mathbb{1}_{h(X)\in R}$, where h is X-measurable
- we call h the *test statistic* and R the *critical region*
- **Important:** *presumed innocent until proven guilty:* reject the null only if enough evidence is collected
- we have to be conservative in choosing H_0

Type I and II errors, ctd.

- type I error = wrongly rejecting the null = **false positive**
- type II error = not rejecting a false null hypothesis = **false negative**

Error types		Null hypothesis is	
Decision		True	False
about H_0	don't reject	correct inference = true negative	type II error = false negative
	reject	type I error = false positive	correct inference = true positive

- think about testing for a disease:
  - **positive** means sick
  - **negative** means healthy
- **Important:** the situation is not symmetric!, generally we want to control the type II error

Building tests from confidence intervals

- **Idea:** from any confidence interval, we can build a test of fit
- suppose that $\hat{C}$ is a $1-\alpha$ level confidence interval for θ
- then in order to test
$$H_0 : \theta = \theta_0 \quad \text{vs} \quad H_1 : \theta \neq \theta_0$$
we can use the test
$$\phi(X) = \mathbb{1}_{\theta_0 \notin \hat{C}} \, .$$
- **What is the level of that test?**
- let $\theta \in \Theta_0$. By definition
$$\begin{aligned} \alpha^\star &= \mathbb{P}_{\theta_0}\left(\phi(X) = 1\right) \\ &= \mathbb{P}_{\theta_0}\left(\theta_0 \notin \hat{C}\right) \\ \alpha^\star &\leq \alpha \end{aligned}$$

One sample Student t-test

- $X_1, \ldots, X_n$ i.i.d. $\mathcal{N}\left(\mu, \sigma^2\right)$, μ and σ unknown
- we want to test

$$H_0 : \mu = \mu_0 \quad \text{vs} \quad H_1 : \mu \neq \mu_0 \,.$$

- **Claim:**

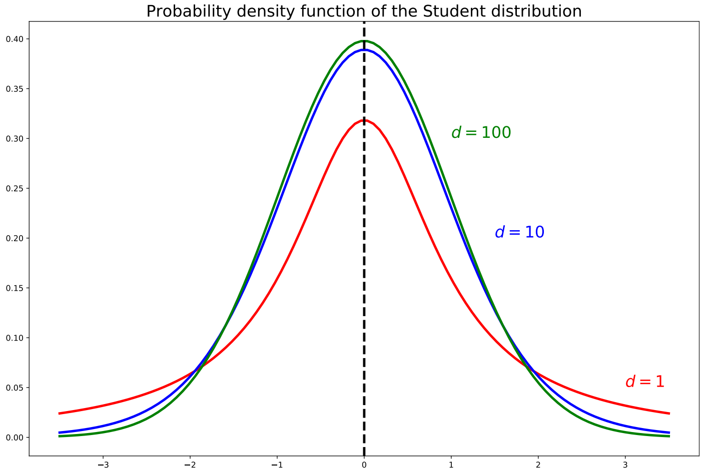
$$T = \frac{\overline{x}_n - \mu}{\hat{\sigma}_n/\sqrt{n}} \sim \mathcal{T}_{n-1} \,,$$

  where $\mathcal{T}_{n-1}$ is the **Student's law** with $n-1$ degrees of freedom
- for any given $\alpha \in (0,1)$, we obtained the $1-\alpha$ level confidence interval for μ

$$\hat{C}_{1-\alpha} = \left[\hat{\mu}_{1,n} - z_{\alpha/2,n-1}\frac{\hat{\sigma}_n}{\sqrt{n}}, \hat{\mu}_{1,n} + z_{\alpha/2,n-1}\frac{\hat{\sigma}_n}{\sqrt{n}}\right]$$

- $\Rightarrow$ the test $\phi(X) = \mathbb{1}_{\mu_0 \notin \hat{C}_{1-\alpha}}$ has level α

Student distribution



Conclusion

- **Summary:**
  - given annotated examples, TCAV provides **class-concept association**
  - quantitatively, for each example, gives a **score** S_C
  - > 0 if the concept is active, < 0 otherwise
  - for a set of examples, an **agglomerated score** TCAV
  - > 0.5 if positive influence, < 0.5 otherwise
- influential work, many extensions
- also used as a ranking tool in other unrelated methods (concrete example in the next section)