8. Concept-based Explainable AI

About the exam

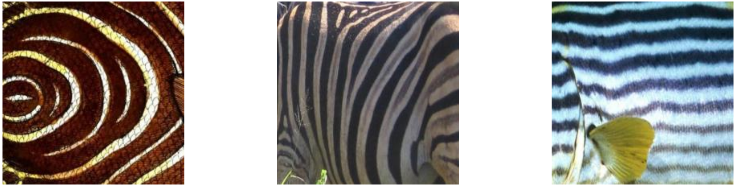
- February 21, 4pm-6pm, 90mn, pen and paper, no documents allowed
- potential questions:
  - What is the difference between the definitions of interpretability and explainability as given in the lecture?
  - Provide an example of a model that is interpretable-by-design. Why is this model considered interpretable?
  - What is a post-hoc method? Give an example.
  - Is MDI (Mean Decrease Impurity) a global or local interpretability method?
  - Give the pseudo-code of LIME for image data.
  - In LIME for images, what criticism can you give from turning pixels black when creating perturbed images? What alternative approaches are available?
  - Using your notation, give the formula for integrated gradients.
  - ...

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

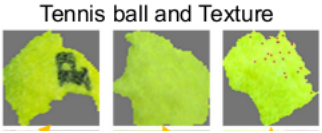


- **Figure:** images corresponding to the `striped` concept from from the `Broden`[80] dataset

[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

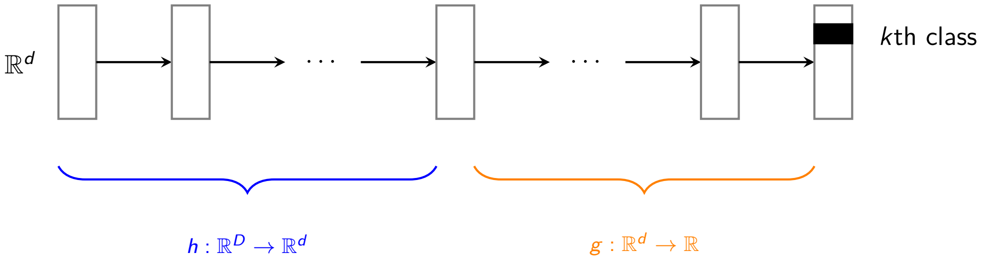
A typology of concept-based XAI

- **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

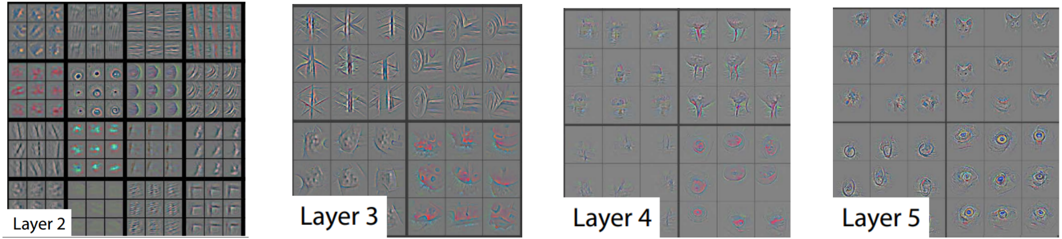
More on latent representation

- **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
- $\mathbb{P}\left(y = 1 \,|\, x\right) = \phi(w^\top x)$ and $\mathbb{P}\left(y = 0 \,|\, x\right) = 1 - \phi(w^\top x)$

Logistic function

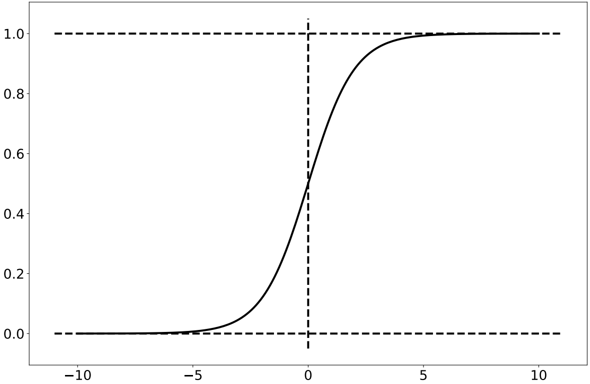
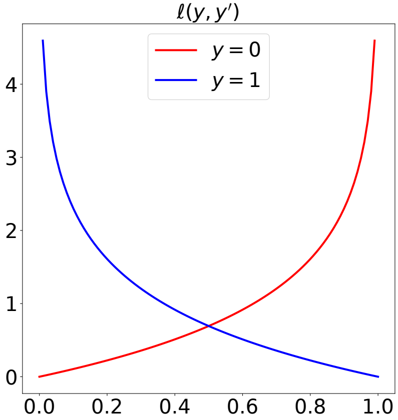


Figure: the logistic function $\phi : t \mapsto 1/(1 + \mathrm{e}^{-t})$.

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 $w \in \mathbb{R}^d$

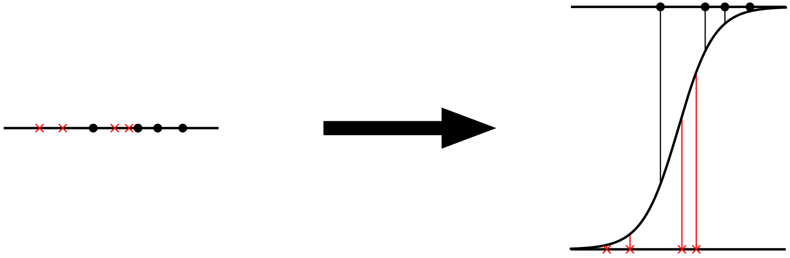
$$\hat{\mathcal{R}}(w) = \sum_{i=1}^{n} \left\{ -(1-y_i)\log(1-\phi(w^\top x_i)) - y_i \log \phi(w^\top x_i) \right\} .$$

- **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:

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

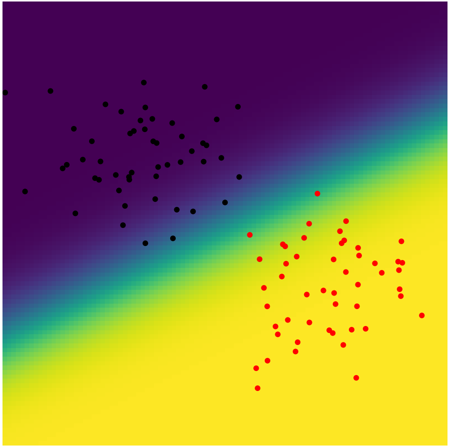
Logistic regression in dimension 1

- **Example:** in dimension one:



Logistic regression in dimension 2

- **Example:** in dimension two:



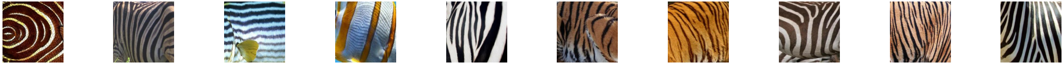
Recap

- **What happens when we call** `sklearn.linear_model.LogisticRegression`?
- `penalty` is $\ell_2 \rightarrow$ 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 $j \in [m]$

$h($$)$ $h($$)$

$h($$)$

$h($$)$ $h($$)$ $h($$)$

$h($$)$

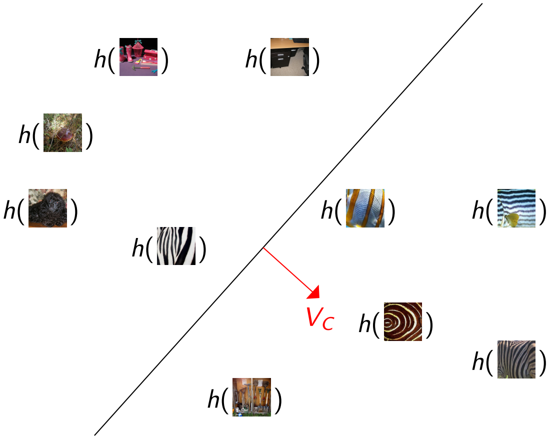
$h($$)$

$h($$)$

$h($$)$

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:**

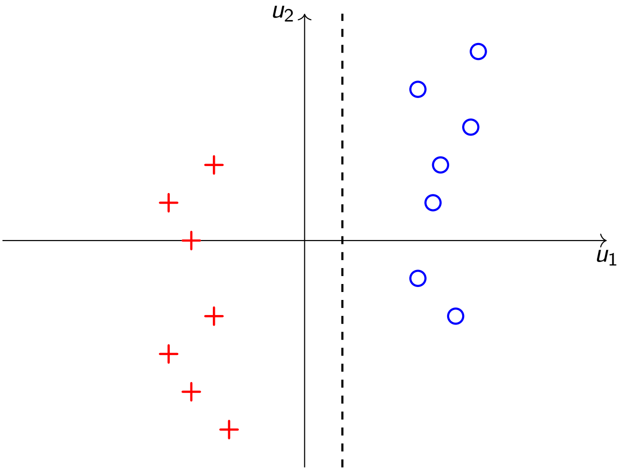
$$\xi =$$  $$\Rightarrow \qquad S_C(\xi) = 0.98\,.$$

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

CAV: intuition

- let us build some intuition under simplifying assumption
- **Assumption (i):** working in the last layer
- $\Rightarrow g(u) = w^\top u$ with $w \in \mathbb{R}^d$
- for any $j \in [d]$, w_j measure exactly the contribution of $h(x)_j$ to the class logit
- we compute:
$$\nabla_{h(x)} g = (w^\top)^\top = w \in \mathbb{R}^d \, .$$
- **Assumption (ii):** concepts and random examples are separated by the hyperplane $e_1^\perp$
- $\Rightarrow$ the concept vector is given by:
$$V_C = \lambda e_1 \, ,$$
with λ a positive constant.
- **Notation:** only true up to complications coming from sampling and optimization

CAV: intuition, ctd.



CAV: intuition, ctd.

- under Assumptions (i) and (ii), we can compute $S(x)$:

$$S(x) = \langle \nabla_{h(x)}, V_C \rangle = w^\top \lambda e_1 = \lambda w_1 \,.$$

- remember: $\lambda > 0$
- thus $\operatorname{sign}(S) = \operatorname{sign}(w_1)$
- $S > 0$ means that $w_1 > 0$: pushing in the direction of e_1 **increases** the class logit
- we can keep this intuition in a more general setting: pushing in the direction of V_C should increase the class logit
- **Why?** Taylor expansion:

$$g(h(x) + \varepsilon V_C) \approx g(h(x)) + (\varepsilon V_C)^\top \nabla_{h(x)} g = f(x) + \varepsilon S(x) \,.$$

- **Disclaimer:** no absolute certainty

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

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

  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	don't reject	correct inference = true negative	type II error = false negative
H_0	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

One sample Student t-test

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

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

- **Claim:**

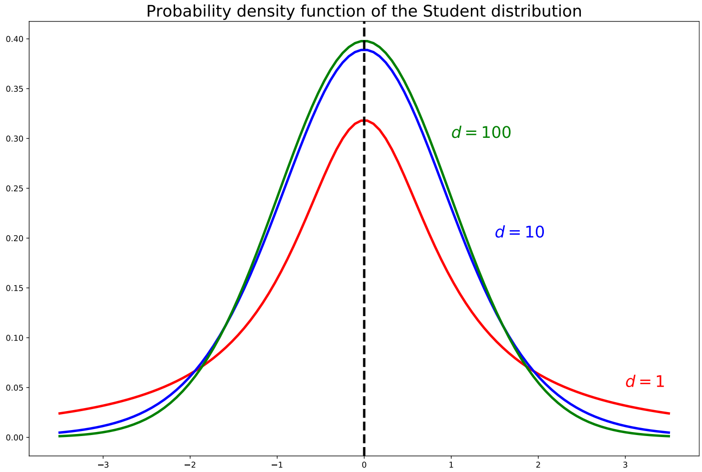
$$T = \frac{\overline{x}_n - \mu}{\hat{\sigma}_n / \sqrt{n}} \sim \mathcal{T}_{n-1} \, ,$$

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

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

- the t-test is given by $\phi(X) = \mathbb{1}_{\mu_0 \notin \hat{C}_{1-\alpha}}$

Student distribution



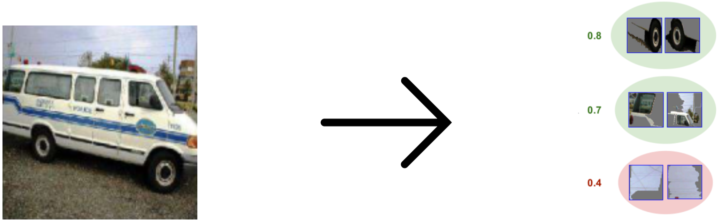
Conclusion

- **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)

8.2. Automatic Concept-based Explanations

Automatic Concept-based Explanations (ACE)

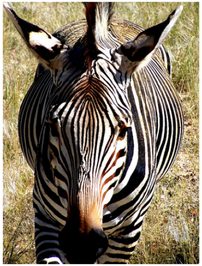
- we now move to another method: ACE[87]
- this method is *unsupervised*, no need for annotated concept images!
- **Big picture:**
  1. start from set of images of the same class;
  2. segment and resize the images;
  3. cluster in the latent space;
  4. remove outliers and rank by TCAV score.
- output concepts are the **clusters**

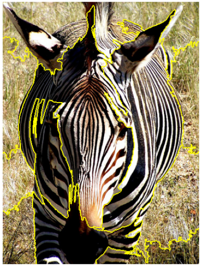


[87] Ghorbani et al., *Towards Automatic Concept-based Explanations*, NeurIPS, 2019

Image segmentation: reminder

- **Overall idea:** group pixels of the image by similar color / texture
- group of pixels = superpixel
- ACE uses SLIC[88] (LIME is using quickshift)





- **Figure:** segmentating a zebra image using SLIC

[88] Achanta et al., *SLIC Superpixels Compared to State-of-the-Art Superpixel Methods*, IEEE TPAMI, 2012

SLIC

- **Executive summary:**
  - map each pixel to $\mathbb{R}^5$ (3 coordinates for color, 2 for position)
  - perform clustering on this set of points
- SLIC uses a variant of k-means[89]
- the **distance** used for clustering is

$$d(i,j)^2 = \frac{d_c^2}{N_c^2} + \frac{d_s^2}{N_s^2},$$

  where d_c (resp. d_s) is the distance in the color (resp. spatial) space, and N_c (resp. N_s) are normalization constants
- **Remark:** connectivity is not enforced

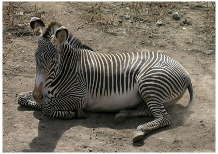
[89] Steinhaus, *Sur la division des corps matériels en parties*, Bull. Acad. Polon. Sci., 1957

ACE step 1: starting images

- let us go back to ACE
- we start with **images from the same class**:



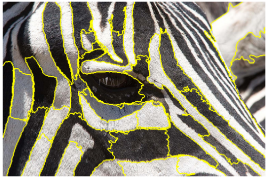




ACE step 2: segment

- each image is **segmented at different scales** using SLIC:

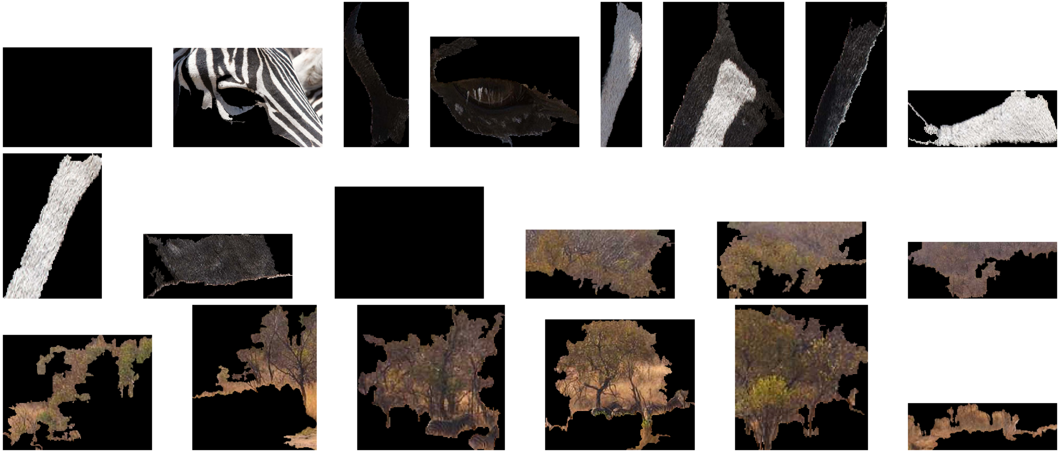




- default scales $= \{15, 50, 80\}$
- **Intuition:** capture all possible concepts (no *a priori* size)
- **Remark:** this step can be replaced by human intervention

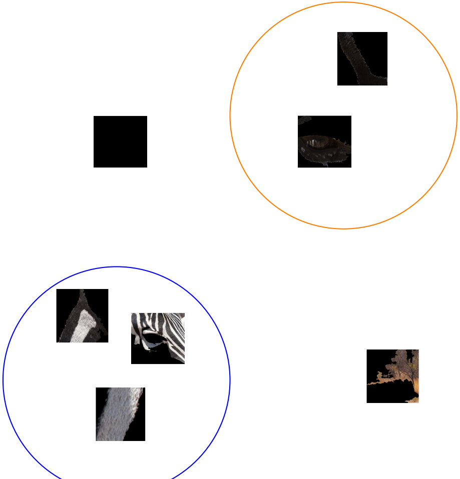
ACE step 2: segment

- extract, crop and pad each segment (gray replacement):



ACE step 3: cluster

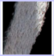
- create clusters in $\mathbb{R}^d$
- remove outliers (keep only 40 points closer to the cluster's center)



ACE step 4: importance score

- rank clusters by TCAV score (see previous section!)

TCAV $= 0.8$

TCAV $= 0.1$

- **Remark:** can use any other concept-importance score

Conclusion

- **Summary:**
  - ACE provides **class-concept association** with no supervision
  - relies on a concept-importance score such as TCAV
- influential work, many extensions:
  - invertible concept-based explanation (ICE)[90]
  - concept recursive activation factorization for explainability (CRAFT)[91]
  - ...

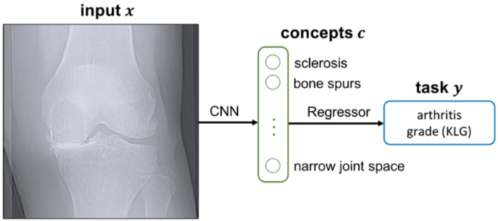
[90] Zhang et al., *Invertible concept-based explanations for CNN models with non-negative concept activation vectors*, AAAI, 2021

[91] Fel et al., *Concept recursive activation factorization for explainability*, CVPR, 2023

8.3. Concept bottleneck models

Introduction

- so far, we have seen *post-hoc* concept-based explanation methods
- both supervised (TCAV) and unsupervised (ACE)
- we now look at ad-hoc methods, starting with **concept bottleneck**[92]
- **Overall idea:** layer dedicated to predicting user-defined concepts
- final output has to rely on this layer $\Rightarrow$ bottleneck
- allows:
  - model transparency (our primary goal)
  - concept intervention
- **Example:**



[92] Koh et al., *Concept bottleneck models*, ICML, 2020

Setting

- **Goal:** predicting $y \in \mathbb{R}$ from input $x \in \mathbb{R}^d$
- assume that we are given training data

$$\{(x^{(1)}, y^{(1)}, c^{(1)}), (x^{(2)}, y^{(2)}, c^{(2)}), \dots, (x^{(n)}, y^{(n)}, c^{(n)})\},$$

  where $X^{(i)}, y^{(i)}$ are as usual, and $c^{(i)} \in \mathbb{R}^k$ are concept vectors
- **Example:** concepts from the arthritis task: `sclerosis`, `bone spurs`, ...
- $c^{(i)} = (10, 0.1, -0.3, \dots)^\top$ corresponds to `sclerosis` being present
- **Concept bottleneck model:** $f(x) = g(h(x))$, where
  - $h : \mathbb{R}^d \to \mathbb{R}^k$ predicts concepts from input
  - $g : \mathbb{R}^k \to \mathbb{R}$ predicts output from concepts

Independent bottleneck

- there are several natural ways to train $f = g \circ h$
- let us call $\hat{g}$ and $\hat{h}$ the trained versions of g and h
- **Loss functions:**
  - $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ for the outputs
  - $\forall j \in [k]$, define $\ell_j : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ the loss for concept j
- **Independent bottleneck:** learn $\hat{g}$ and $\hat{h}$ independently:

$$\hat{h} \in \operatorname*{arg\,min}_h \sum_{i=1}^{n} \sum_{j=1}^{k} \ell_j(h_j(x^{(i)}), c_j^{(i)}) , \qquad \text{and} \qquad \hat{g} \in \operatorname*{arg\,min}_g \sum_{i=1}^{n} \ell(g(c^{(i)}), y^{(i)}) .$$

- **Intuition:** learn (independently) a good concept predictor and a good predictor relying only on concepts
- **Beware:** although $\hat{g}$ trained using true concepts, $\hat{f} = \hat{g} \circ \hat{h}$

Other possibilities

- **Sequential bottleneck:** $\hat{h}$ learned as before, $\hat{g}$ learned using $\hat{h}$
- namely,

$$\hat{h} \in \arg\min_{h} \sum_{i=1}^{n} \sum_{j=1}^{k} \ell_j(h_j(x^{(i)}), c_j^{(i)}) \,, \qquad \text{and} \qquad \hat{g} \in \arg\min_{g} \sum_{i=1}^{n} \ell(g(\hat{h}(x^{(i)})), y^{(i)}) \,.$$

- **Joint bottleneck:** minimize a weighted sum of the two objectives:

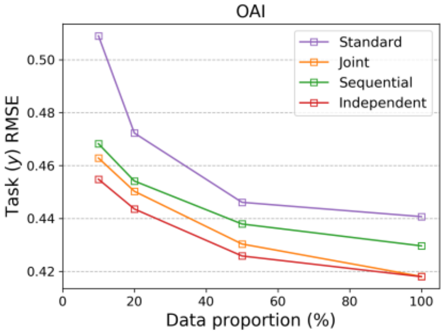
$$\hat{g}, \hat{h} \in \arg\min_{g,h} \sum_{i=1}^{n} \left[\ell(g(h(x^{(i)})), y^{(i)}) + \lambda \sum_{j=1}^{k} \ell_j(h_j(x^{(i)}), c_j^{(i)}) \right] \,,$$

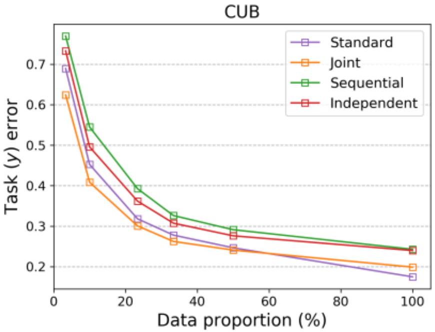
  with $\lambda > 0$ some hyperparameter
- **Standard model:** ignores concepts altogether:

$$\hat{g}, \hat{h} \in \arg\min_{g,h} \sum_{i=1}^{n} \ell(g(h(x^{(i)})), y^{(i)}) \,.$$

Empirical results

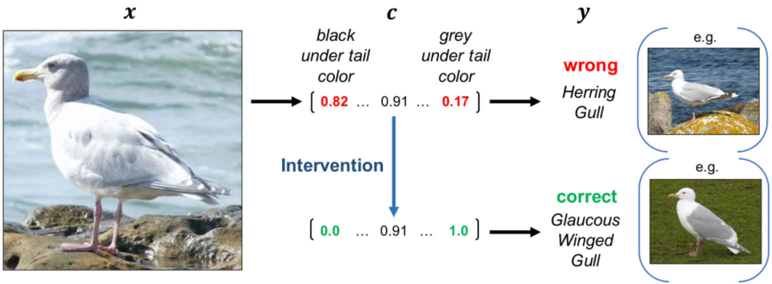
- all models are good at predicting concepts
- then the metric is really accuracy: depends on the task
- … and always a bit smaller than without relying on concepts :(





Concept intervention

- **Concept intervention:** modifying concept values to get more accurate prediction
- **Example:**



Summary

- **Concept bottleneck:** explainable-by-design concept-based model
- requires user-defined concepts
- allows for concept intervention
- many extensions
- **Remark:** also possible to perform transplantation on existing network, introducing concept layer instead of existing layer