Topic 4: Trustworthy Vision

Some Factors Affecting Trust in Deep Learning

- **Models Complexity - Non-Decomposability into Simple Components**
  - Explainability
  - Interpretability
- **Social Discrimination and Data/Model Misrepresentations**
  - Disparate Treatment (e.g. Social Biases in Datasets)
  - Disparate Impact (e.g. Discriminative Outcomes)
- **Unreliable Inference even to Minor Input Disruptions**
  - Adversarial Examples

Impact of Stakeholders on Explainable AI (XAI)

How do diverse stakeholders perceive about neural networks?

- Decision Maker
  - Use predictions as recommendations to make appropriate judgements
  - e.g. doctors trying to diagnose patients
  - Cares about global explanations as well as local explanations
- Affected User
  - Analyze their inputs in retrospect to change the future outcome
  - e.g. patients
  - Cares only about local explanations
- Regulator
  - Ensures the model is safe and compliant with
  - e.g. government official trying to validate the model
  - Cares about both global explanations and local explanations
- Data Scientist
  - Improve model performance
  - e.g. some of you in the future!

Types of Explainable AI (XAI)

Local Explanations: Explain predictions for a given input data point

- Saliency Maps
- Class Activation Maps (CAM)
- Grad-CAM

Global Explanations: Explain the overall model

- ?

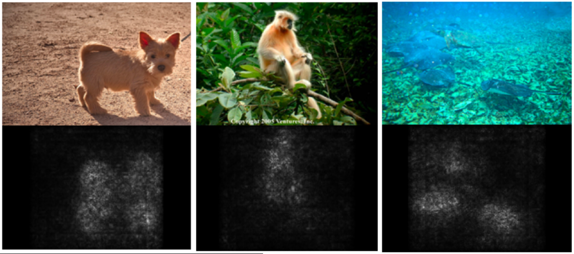
Saliency Maps[1]

- Consider input image I_0 of size $m \times n$, and a class c
- Highly non-linear class score function $S_c(I)$ in deep NNs $\Rightarrow$

  Approximate $S_c(I)$ with a linear function in the neighborbood of I_0 using Taylor's expansion:

$$S_c(I) \approx w^T I + b, \text{ where } w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0} \text{ can be found via backprop.}$$

- Saliency Map $M_{i,j} = |w_{h(i,j)}|$, where $h(i,j)$ is the index in w that corresponds to $(i,j)^{th}$ pixel in I_0.
- Multi-channel images $\Rightarrow M_{i,j} = \max_c |w_{h(i,j,c)}|$
- Also, a regression problem to produce images that maximize a given class score



[1] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," ArXiv:1312.6034, 2013.

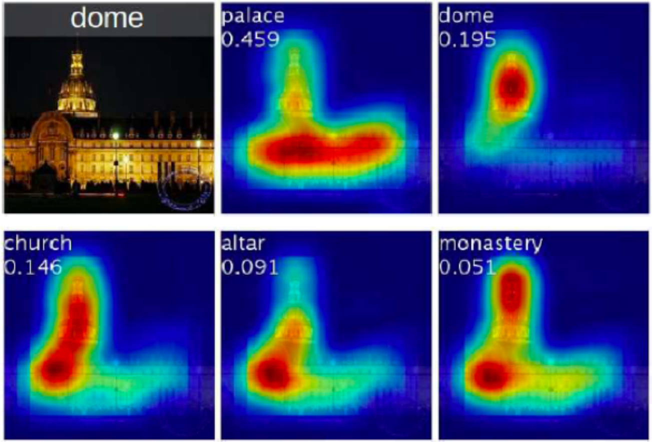
Class Activation Maps (CAM[2])

- Conv. layers are natural object detectors $\Rightarrow$ *Global average pooling* (GAP) instead of FC layers
- Let $f_k(x, y)$ denote activation of unit k at location (x, y).
- Result of GAP at unit k: $F_k = \frac{1}{Z} \sum_{x,y} f_k(x, y)$
- Class score: $S_c = \sum_k w_k^c F_k$ (ignore bias term) $\Rightarrow$ Softmax output: $\dfrac{\exp(S_c)}{\sum_c \exp(S_c)}$
- CAM: $M_c(x, y) = \sum_k w_k^c f_k(x, y) \Rightarrow S_c = \frac{1}{Z} \sum_{x,y} M_c(x, y)$
- Need to retrain the NN for weights w_k^c
- Upscale CAM to input size.

[2] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921-2929, 2016.

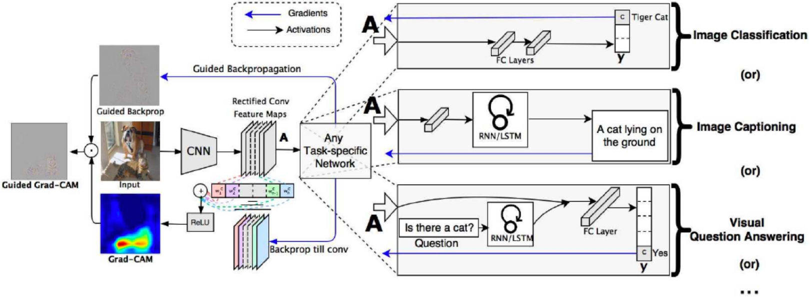
CAM (cont...)

- Example of CAMs generated from top-5 predicted categories
- Note that the *dome* class activates the upper round portion, while *palace* activates the lower flat portion of the compound.



Grad-CAM[3]

- Class score: $S_c = \sum_k w_k^c F_k$, where $F_k = \frac{1}{Z}\sum_{x,y} f_k(x,y)$
- CAM: $M_c(x,y) = \sum_k w_k^c f_k(x,y)$
- $w_k^c = \dfrac{\partial S_c}{\partial F_k} = \dfrac{\partial S_c}{\partial f_k(x,y)} \left(\dfrac{\partial F_k}{\partial f_k(x,y)}\right)^{-1} = Z \cdot \dfrac{\partial S_c}{\partial f_k(x,y)}$
- $\displaystyle\sum_{x,y} w_k^c = Z \cdot \sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)} \quad \Rightarrow \quad w_k^c = \sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)}$ (No need to retrain!)
- Grad-CAM: $\tilde{M}_c(x,y) = ReLU\left[M_c(x,y)\right]$

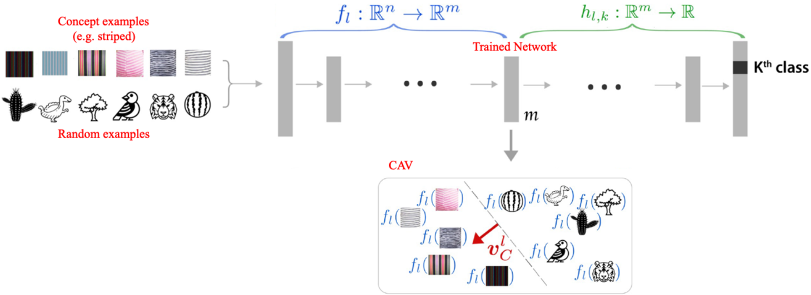


[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618-626, 2017.

Interpretable AI

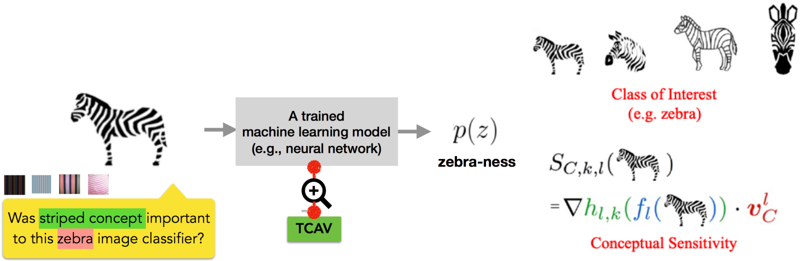
- Concept Activation Vectors (CAV)
- Uncertainty Quantification and Bayesian Neural Networks

Concept Activation Vectors (CAV)



- $f_l(x)$ takes input x and outputs layer l activations $a \in \mathbb{R}^M$.
- $h_{l,k}(a)$ takes layer l activation a and outputs the class-k logit $\in \mathbb{R}$.
- Given a user-defined concept C, let
  - P_C denote the set of images that positively represent the concept
  - N_C denote the set of images that negatively represent the concept
  - $A_P = \{f_l(x) | x \in P_C\}$, $A_N = \{f_l(x) | x \in N_C\}$
- Train a linear classifier to find a hyperplane with normal $v_C^l \in \mathbb{R}^M$ (CAV) that separates A_P and A_N.

Testing with CAV (TCAV[4])

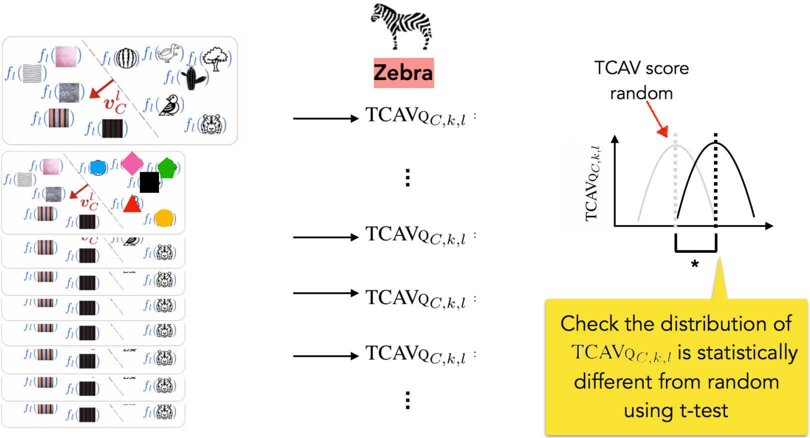


- **Conceptual Sensitivity:** A directional derivative $S_{C,k,l}(x)$ that measures the sensitivity of logit output to change in CAV.
- In saliency maps, we compute the gradient wrt input pixels instead.
- **TCAV:** Aggregate per-input conceptual sensitivity over a class k

$$TCAV_{C,k,l} = \frac{|\{x \in \mathcal{X}_k | S_{C,k,l}(x) > 0\}|}{|\mathcal{X}_k|},$$ where $\mathcal{X}_k$ denotes all inputs for class k.

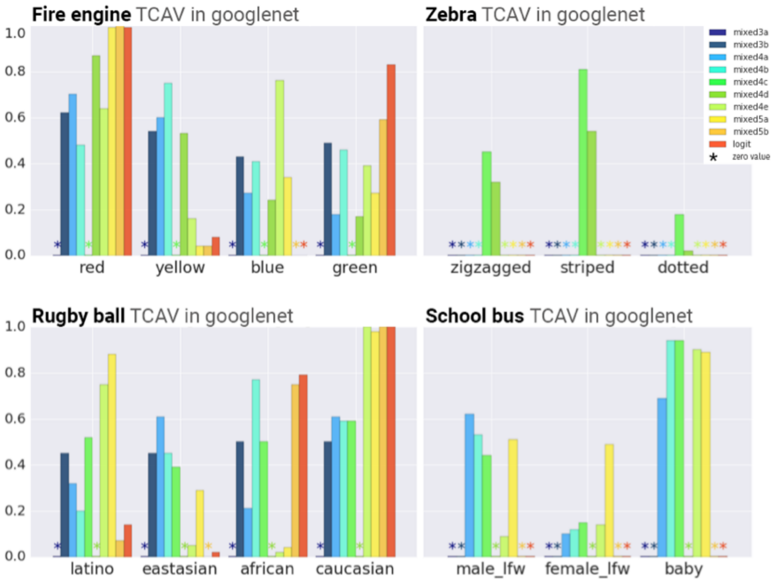
[4] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, and F. Viegas, “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV),” In *International Conference on Machine Learning (ICML)*, pp. 2668-2677, 2018.

Statistical Significance of CAV

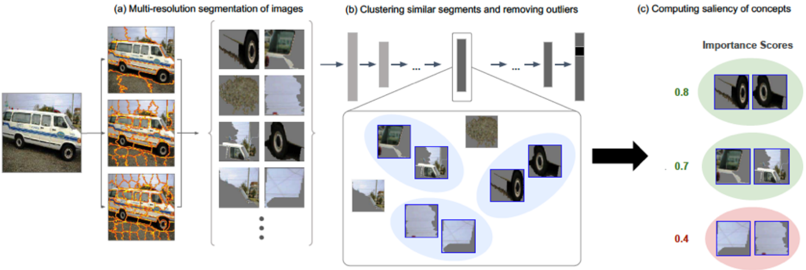


- Note: TCAV is very sensitive to low-quality random CAV.
- Compute TCAVs T times using different N_C sets to obtain $\{TCAV_{C,k,l}^{(i)}\}_{i=1}^{T}$
- Perform two-sided t-test.

Example: TCAV on GoogLeNet



Automatic Concept-Based Explanations (ACE[5])



Desired Properties of Concept-Based Explanation:

- ***Meaningfulness:*** Examples need to be semantically meaningful on its own. Also, multiple individuals should associate similar meaning to the same concept. (e.g. a group of pixels that contains a specific texture/object)
- ***Coherency:*** xamples need to be perceptually similar to each other, but also different from examples of other concepts.
- ***Importance:*** The concept's presence is necessary for the true prediction

[5] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. "Towards Automatic Concept-Based Explanations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

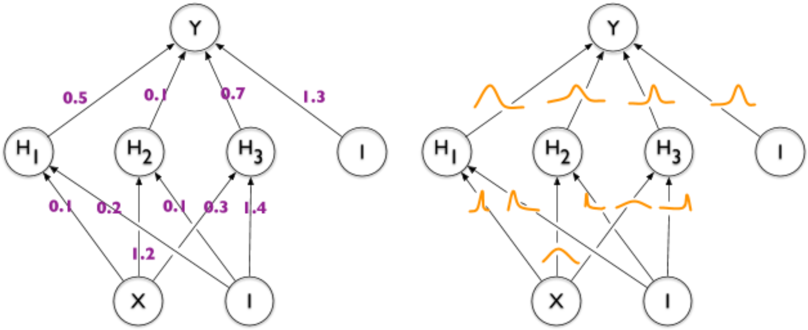
Uncertainty Quantification

Two types of uncertainty:

- **Aleatoric Uncertainty:** Confidence in input data
  - High when input data is noisy
  - Cannot be reduced by adding more data
  - Can be estimated using likelihood methods using neural networks
- **Epistemic Uncertainty:** Confidence in Prediction
  - High when training data is small
  - Can be reduced by adding more data
  - Very difficult to estimate (Knowing when the model does not know the answer)

Solution to Epistemic Uncertainty: Bayesian Neural Networks

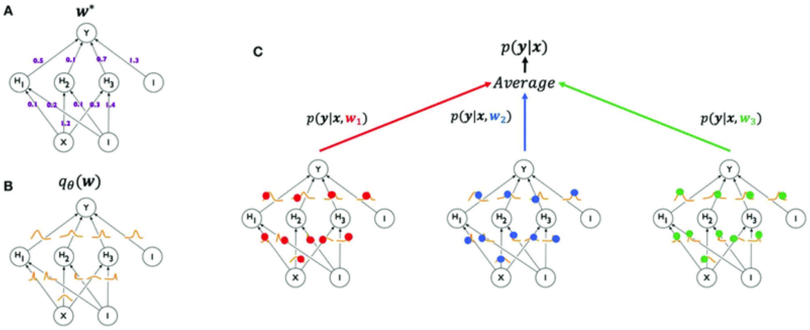
Bayesian Neural Networks (BNNs)



- Train weight distributions, as opposed to just weights as in traditional NNs.
- Assume a prior distribution for weights $p(\mathbb{W})$, and a dataset $(\boldsymbol{X}, \boldsymbol{Y})$.
- Use Bayes' rule to update weight distribution via computing its posterior:

$$p(\mathbb{W}|\boldsymbol{X}, \boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{W}) \cdot p(\mathbb{W})}{p(\boldsymbol{Y}|\boldsymbol{X})}$$

Emulating BNNs through Monte-Carlo Sampling[7]



- Sample weights from the trained distribution of weights several times
- Compute the average logit probability at the output of each class
- Similar approaches: Use Dropout[6] in Testing Phase to capture epistemic uncertainty

---

[6] Y. Gal, and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," In *International Conference on Machine Learning (ICML)*, pp. 1050-1059, PMLR, 2016.

[7] B. Lakshminarayanan, A. Pritzel, C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Advances of Neural Information Processing Systems (NeurIPS)*, 2017.