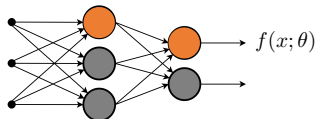


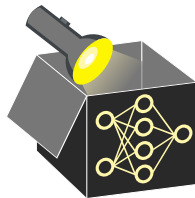
Interpretable Machine Learning

Visualizing Neural Networks



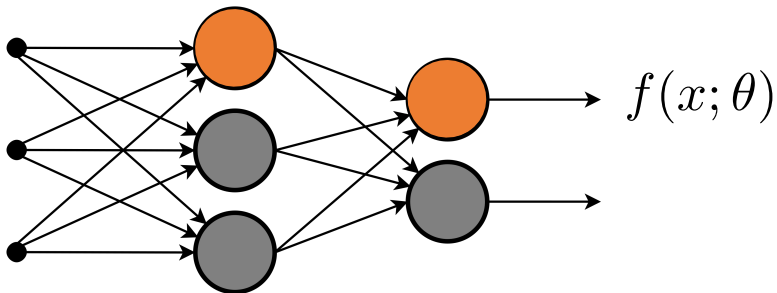
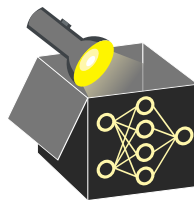
Learning goals

- Visualizing architectural units
- Visualizing filters in CNNs
- Visualizing attention maps



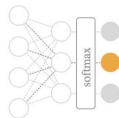
INSPECTING THE MODEL UNITS

- Neural Networks architectural units can be inspected to provide insights
- What happens to the input signal as it travels through the network ?
 - Activations: Activation in neural networks are sparse
 - Attention units: Encode the importance of input representation units

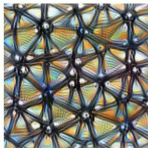


VISUALIZING NEURAL NETWORK ARCHITECTURAL UNITS

- Search for examples where individual features have high values —
 - Either for a neuron at an individual position, or for an entire channel



Neuron



Channel



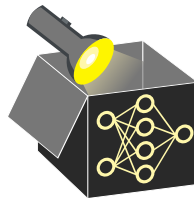
Layer/DeepDream



Class Logits

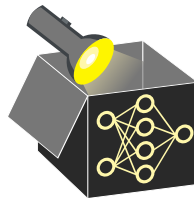
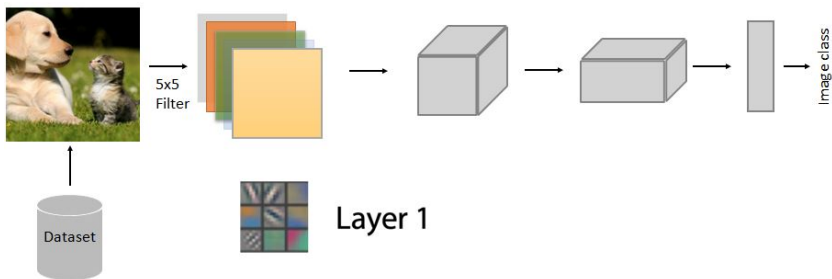


Class Probability



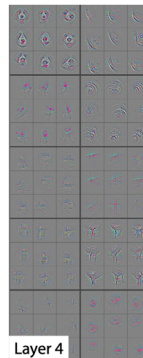
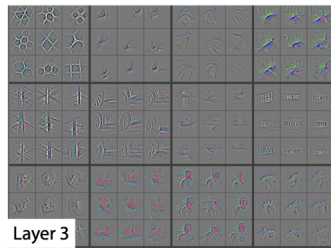
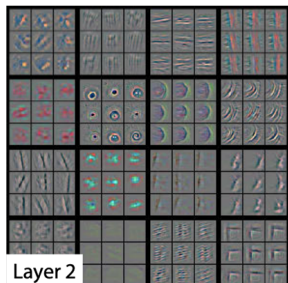
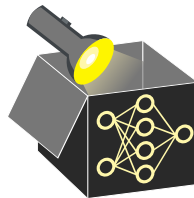
VISUALIZING FILTERS IN A CNN

- Most of the aggregated values at neurons do not result in activations
- Find image patches in dataset that maximally activate/excite a unit

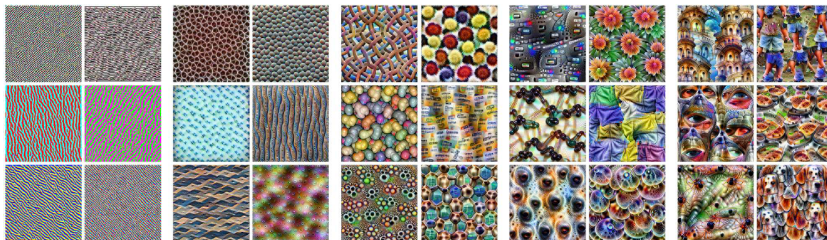


FEATURE EXTRACTION EVOLUTION

- Lower layers extract lower-level features
- Higher layers compose extracted features to compose high-level features



LAYERWISE VISUALISATION OF CNNs



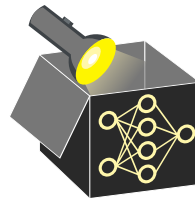
Edges (layer conv2d0)

Textures (layer mixed3a)

Patterns (layer mixed4a)

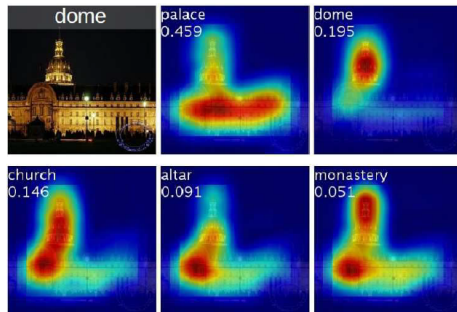
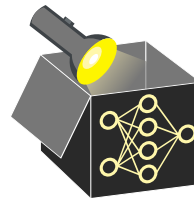
Parts (layers mixed4b & mixed4c)

Objects (layers mixed4d & mixed4e)



CLASS ACTIVATION MAPS

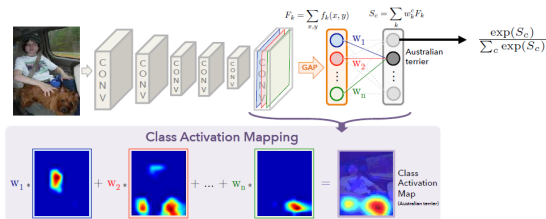
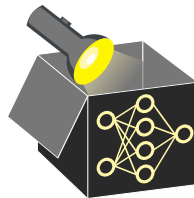
- CAMs are specific to CNNs
- Class activation map or CAM highlights class-specific discriminative regions
 - Different classes induce different activations



CLASS ACTIVATION MAPS

- Let the activation at unit k , at the location (x,y) in the last layer - $f_k(x, y)$
- Global avg. pooling at unit k - $F_k = \sum_{x,y} f_k(x, y)$
- For a given class

$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}, \quad S_c = \sum_k w_k^c F_k$$

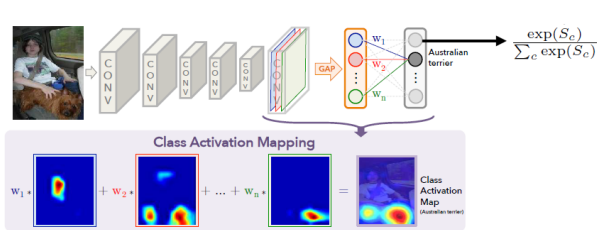


CLASS ACTIVATION MAPS

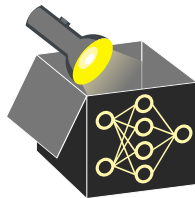
- Input: Take a pre-trained CNN model
- Output: weight vectors for each classes
- How do we learn the weights?
 - Average pooling of the feature maps in the last layer

$$S_c = \sum_k w_k^c F_k$$

- Weights learned using simple logistic regression

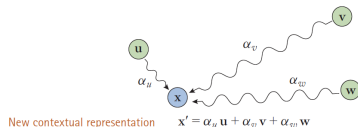
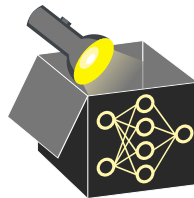


$$\frac{f_k(x, y)}{\sum_{x, y} f_k(x, y)}$$



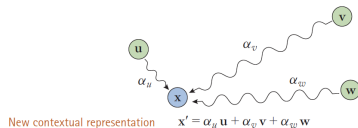
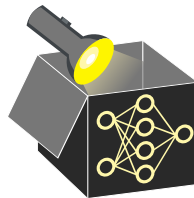
ATTENTION IN LANGUAGE

- Attention mechanism in neural language models is crucial for extracting latent features
- Self-attention in language is aimed at re-representing the initial representation based on the context
- Neural models consume non-contextual token-level representations and output contextual token-level representation



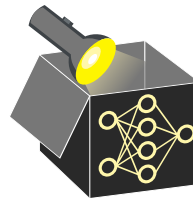
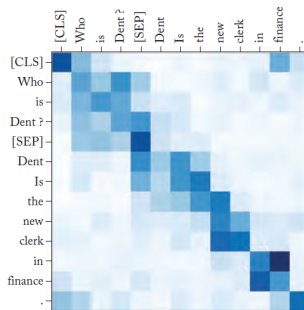
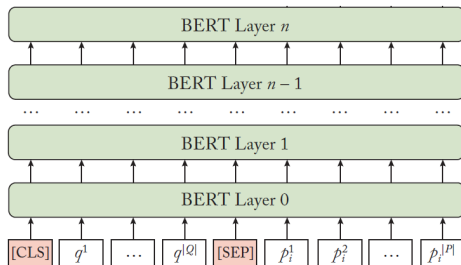
ATTENTION IN LANGUAGE

- Attention mechanism in neural language models is crucial for extracting latent features
- Self-attention in language is aimed at re-representing the initial representation based on the context
- Neural models consume non-contextual token-level representations and output contextual token-level representation

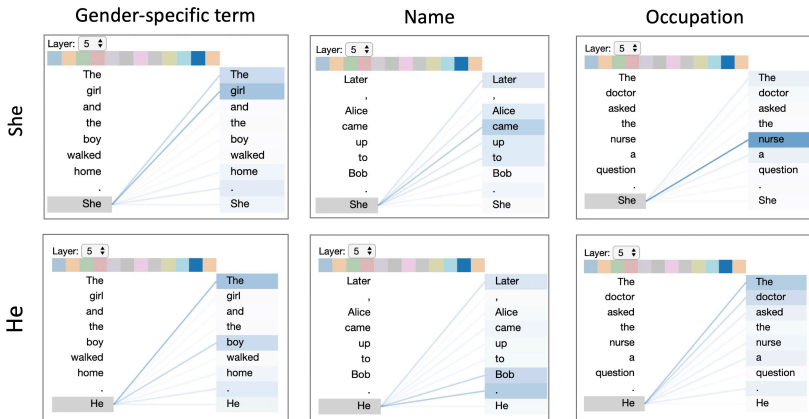
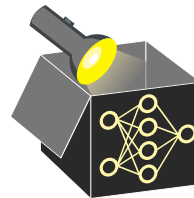


$$\alpha_u = \frac{e^{\text{sim}(u,x)}}{e^{\text{sim}(u,x)} + e^{\text{sim}(v,x)} + e^{\text{sim}(w,x)}}; \quad \text{sim}(u,x) = x \cdot Wu$$

ATTENTION MAPS IN TRANSFORMERS



VISUALIZING ATTENTION UNITS



OTHER INTERACTIVE VISUALISATIONS

- Interactive visualization by Chris Olah:
<https://distill.pub/2018/building-blocks/>
- <https://distill.pub/2017/feature-visualization/>
- Deep Dream
- De-Convolution
- Visualizations in Language: <https://github.com/jessevig/bertviz>
- ...

