iML: Post-hoc Methods for Neural Networks

Motivation

Marius Lindauer and Avishek Anand

Winter Term 2021

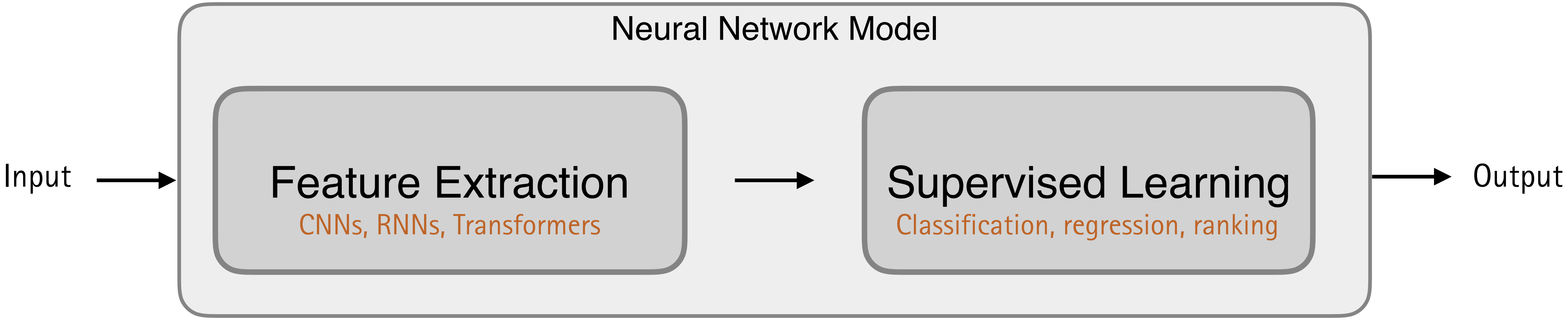
Neural Networks as Complex ML models

Neural Networks as Complex ML models

- Neural networks are over parameterised
  - Vision models and Language models routinely have > millions of params
  - Sometimes #parameters > #input instances
  - Which and how do the features, parameters, training instances contribute towards the final decision ?

Neural Networks as Complex ML models

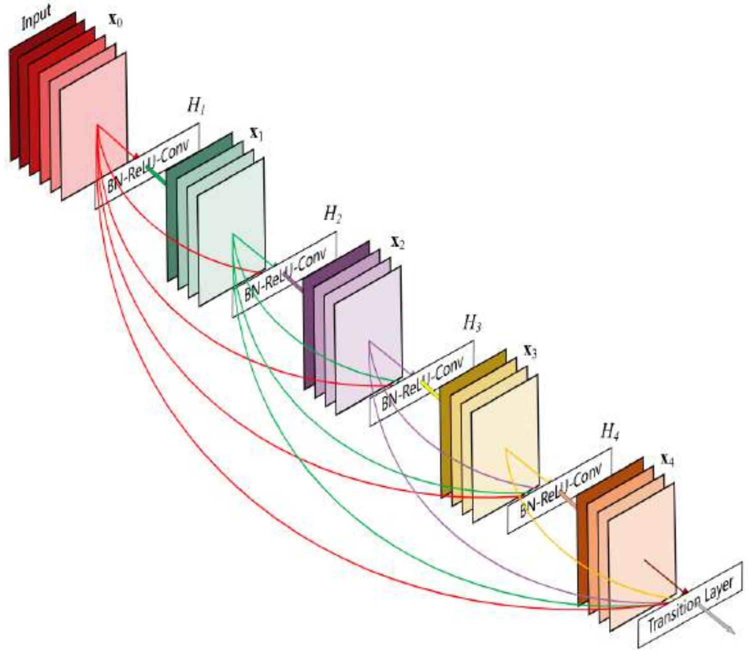
- Neural networks are over parameterised
  - Vision models and Language models routinely have > millions of params
  - Sometimes #parameters > #input instances
  - Which and how do the features, parameters, training instances contribute towards the final decision ?



Neural Networks as Complex ML models

- Neural networks are compositional and non-linear systems
  - The success of neural networks is due to their **depth**
    - Depth results in compositional behaviour
  - Non-linearity between layers helps capture non-linear relationships

- Depth and non-linearity leads to lack of interpretability

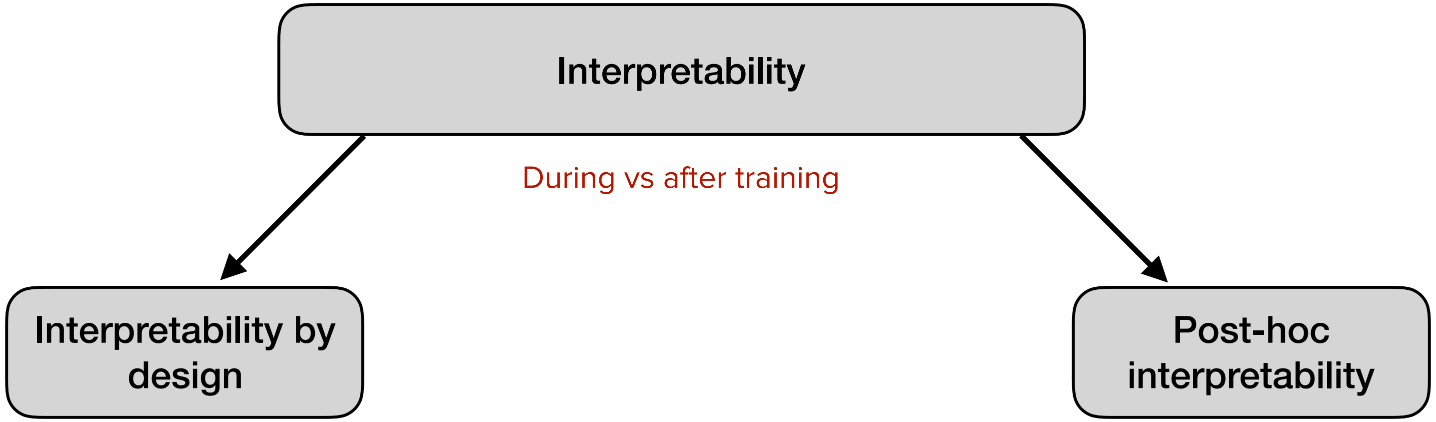


Model-specific interpretability

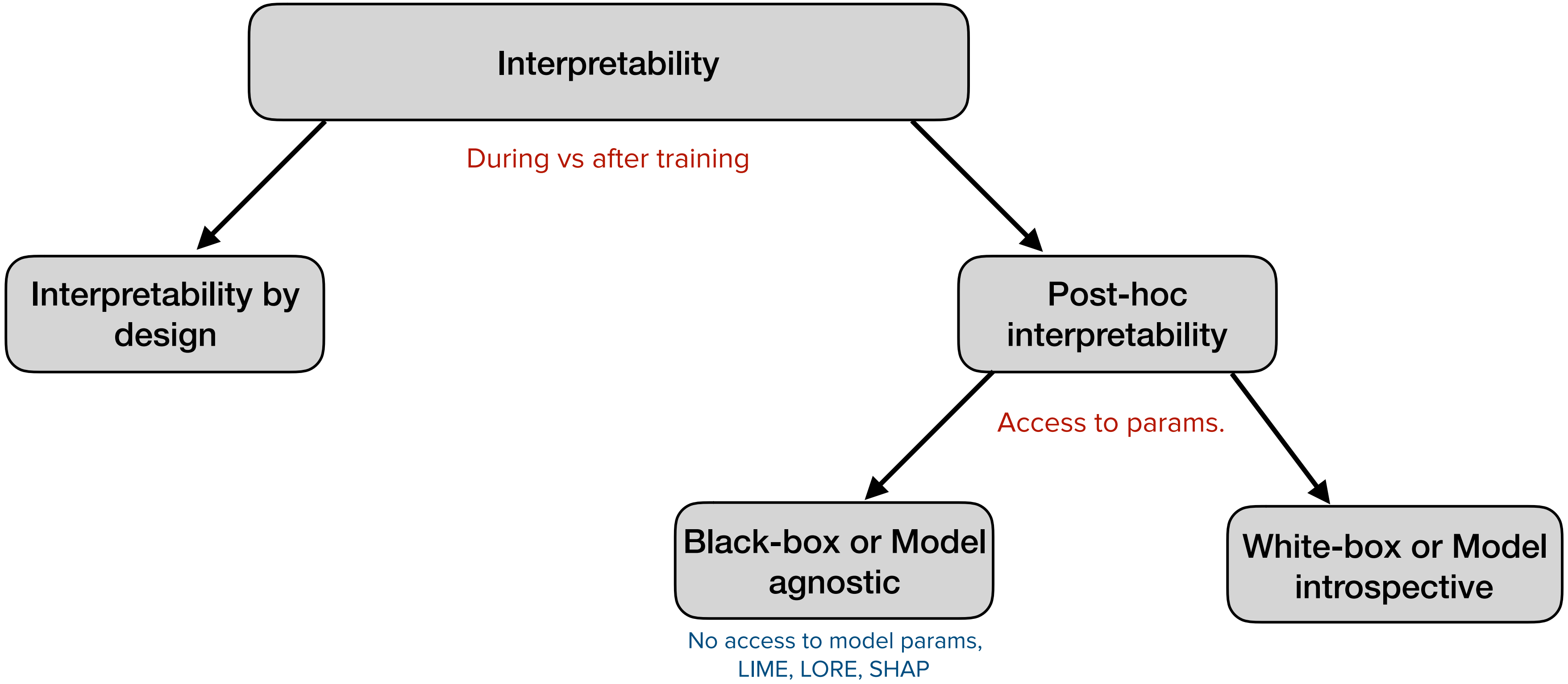
- What types of neural models are out there ?
  - **For vision:** Convolutional Neural Nets
  - **For language, speech:** Recurrent Neural Nets, Transformer Models
  - **For recommendation systems, ranking:** Factorization-based Models, Embeddings models

- Each of the domains have their challenges and have developed specific approaches for interpretability
  - We will focus on first principles that can be applied to most models
  - We will discuss adaptations to each data modality as and when required

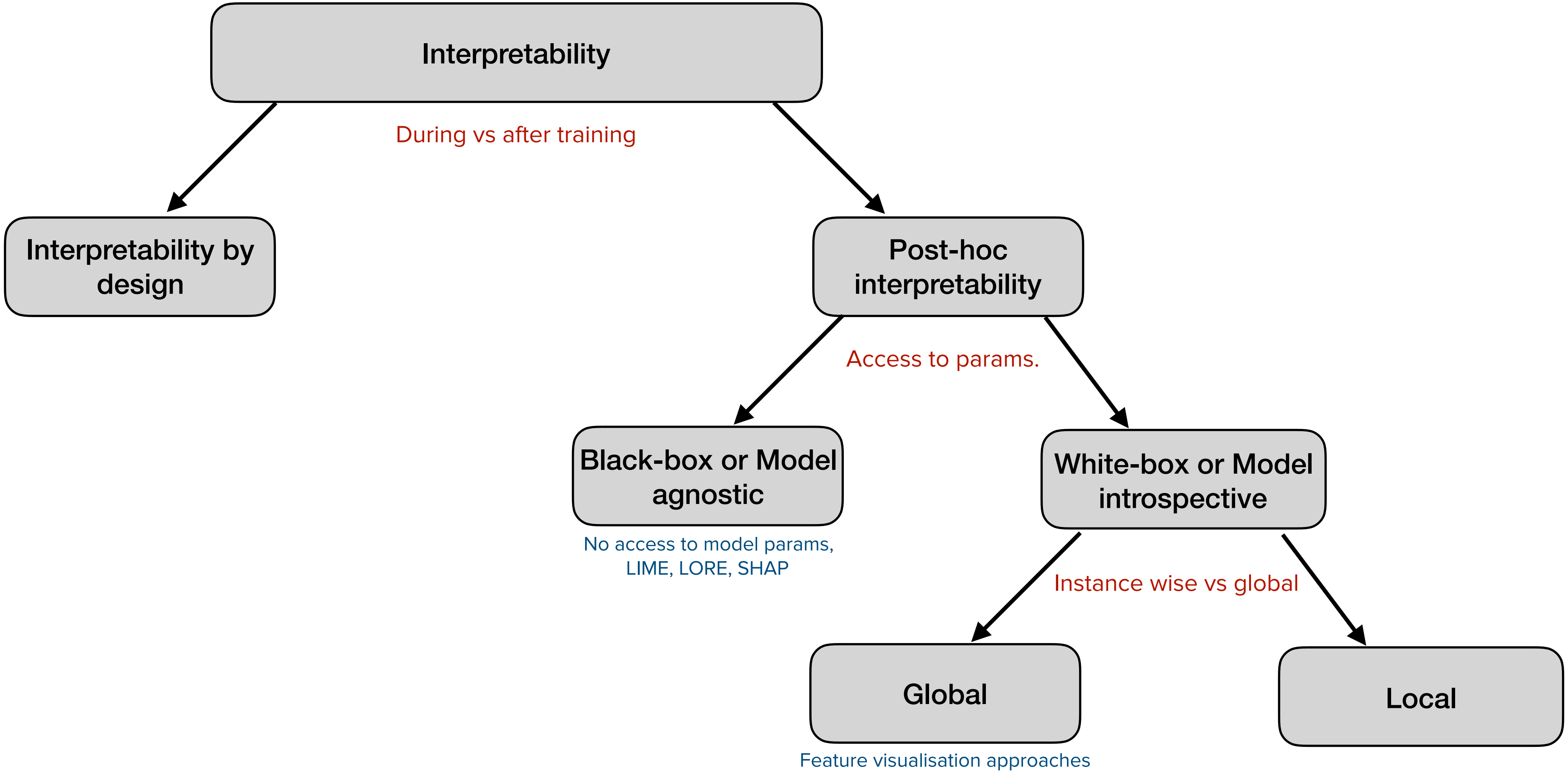
Interpretability landscape in Neural Networks



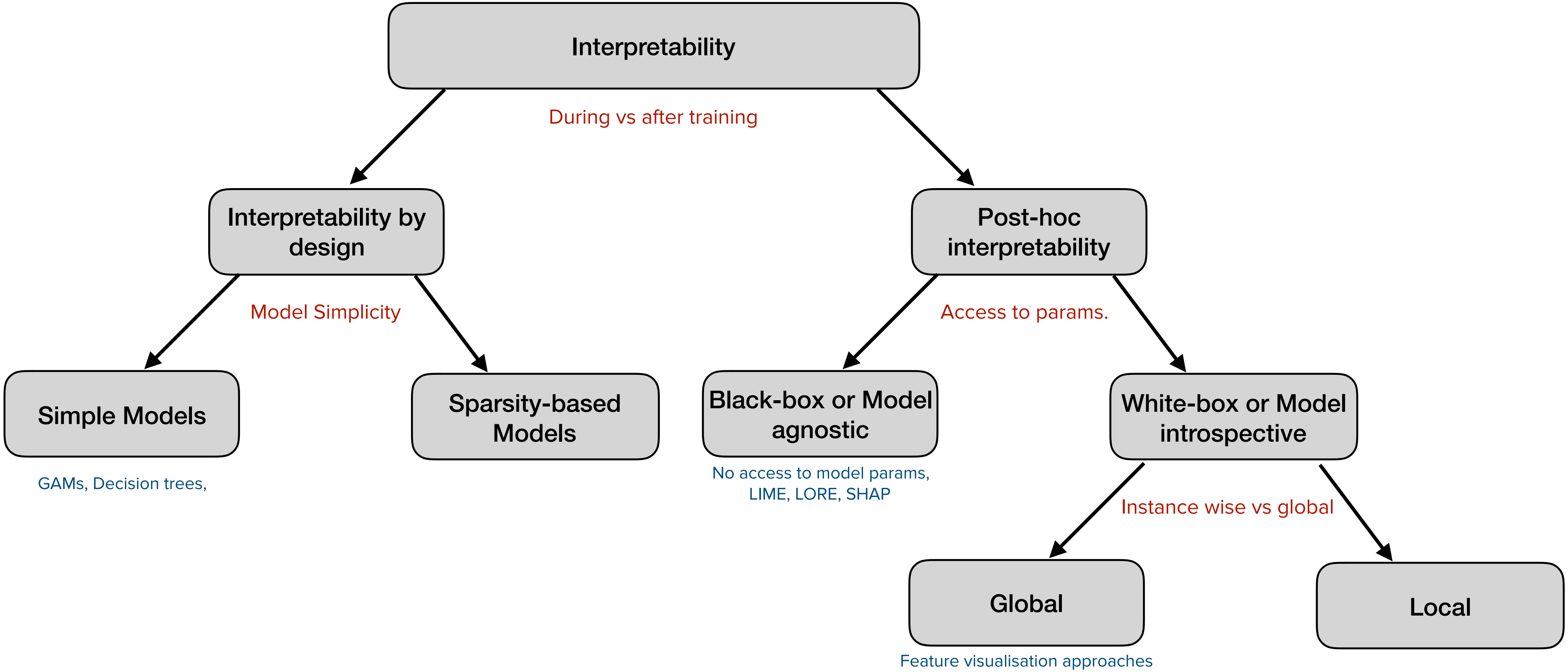
Interpretability landscape in Neural Networks



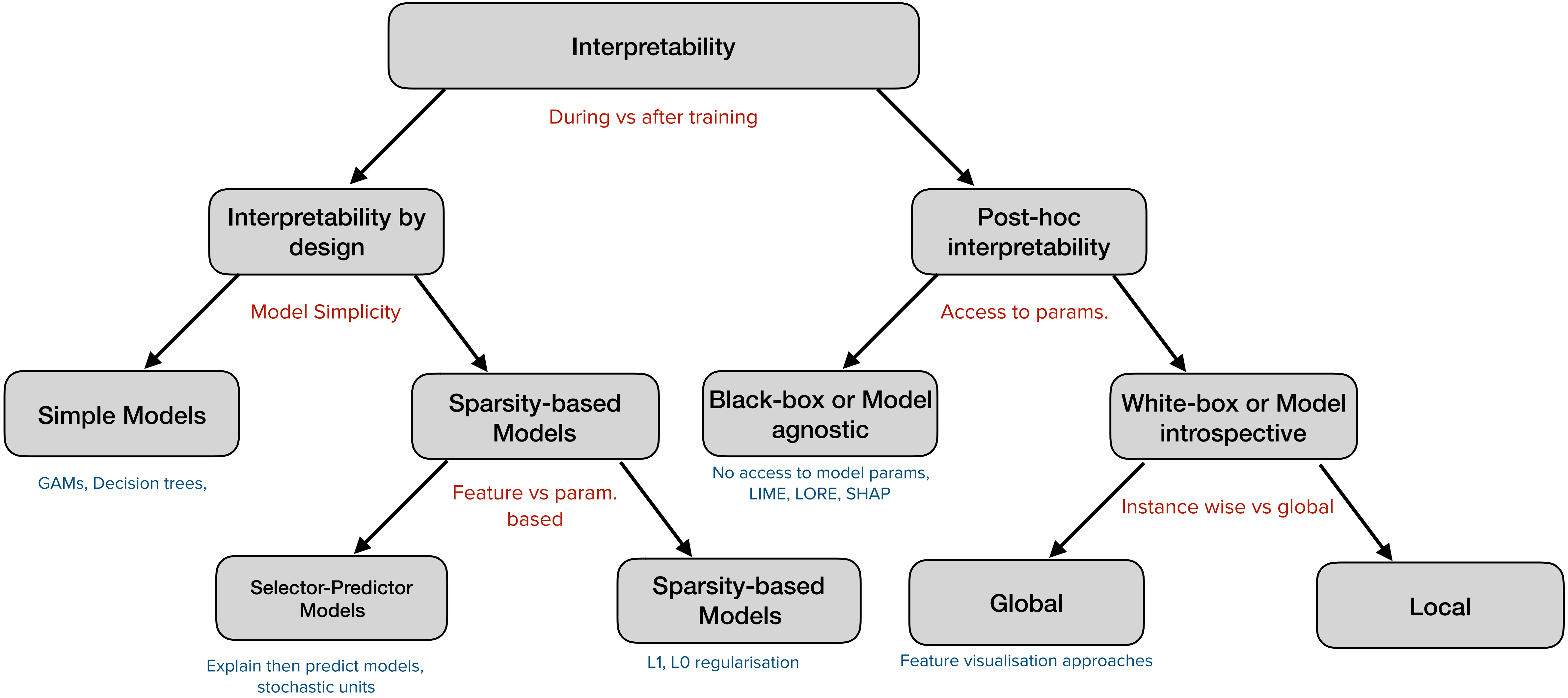
Interpretability landscape in Neural Networks



Interpretability landscape in Neural Networks



Interpretability landscape in Neural Networks



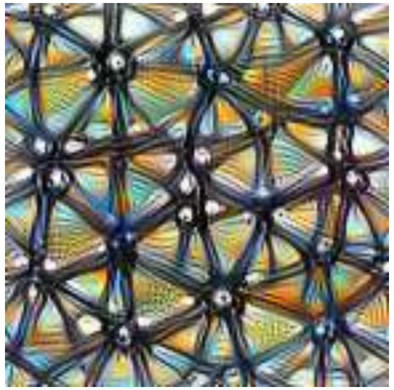
How can we interpret Neural Models ?

- **Feature visualization:** Visualizing components of the neural networks
  - Activations of neurons
  - Attention values
  - Gradient flow

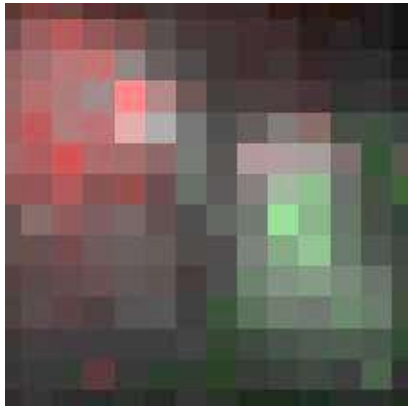
- **Feature attributions:** relevant input features
  - Which input features are responsible for the given decision ?
  - Sensitivity analysis using gradient-based methods
  - Using black-box methods like LIME, SHAP, etc.

How can we interpret Neural Models ?

- **Feature visualization:** Visualizing components of the neural networks



- **Feature attributions:** relevant input features



iML: Post-hoc Methods for Neural Networks

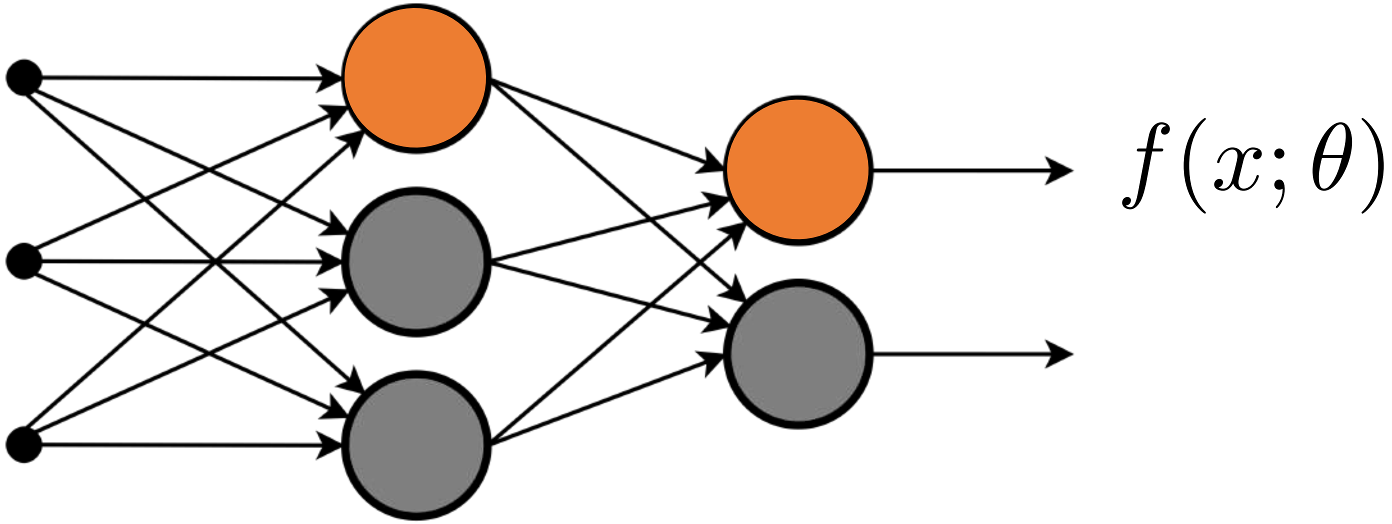
Visualizing Neural Networks

Marius Lindauer and Avishek Anand

Winter Term 2021

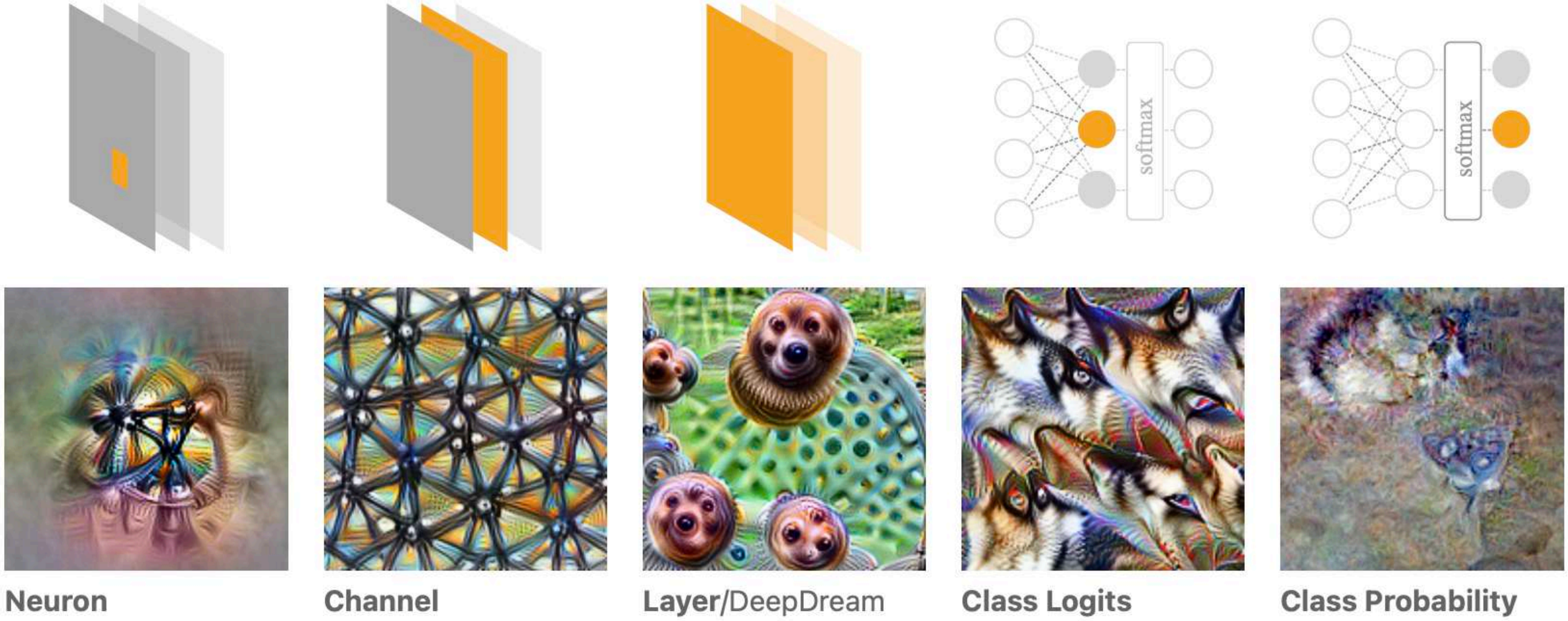
Inspecting the Model units

- 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

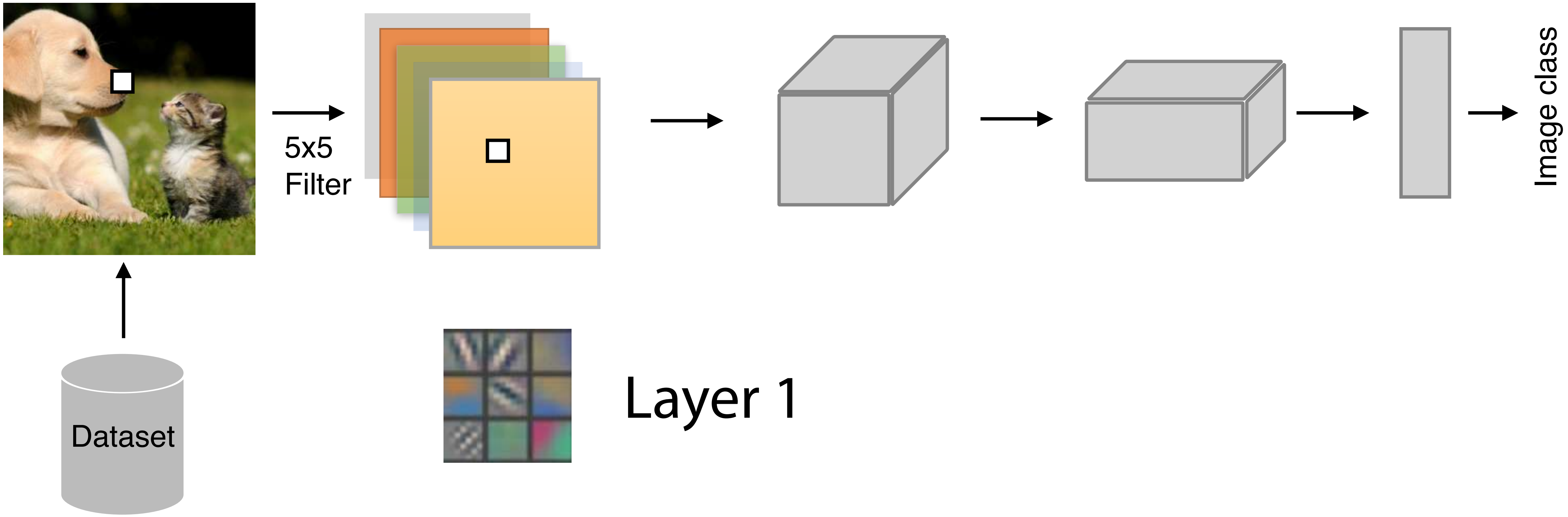
- Search for examples where individual features have high values –
  - Either for a neuron at an individual position, or for an entire channel



https://distill.pub/2017/feature-visualization/

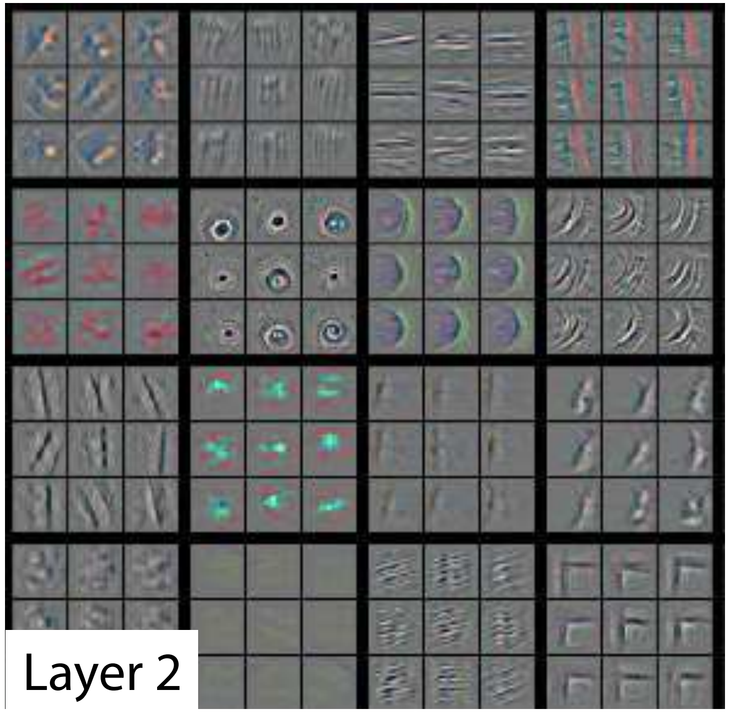
Visualizing Filters in a CNN

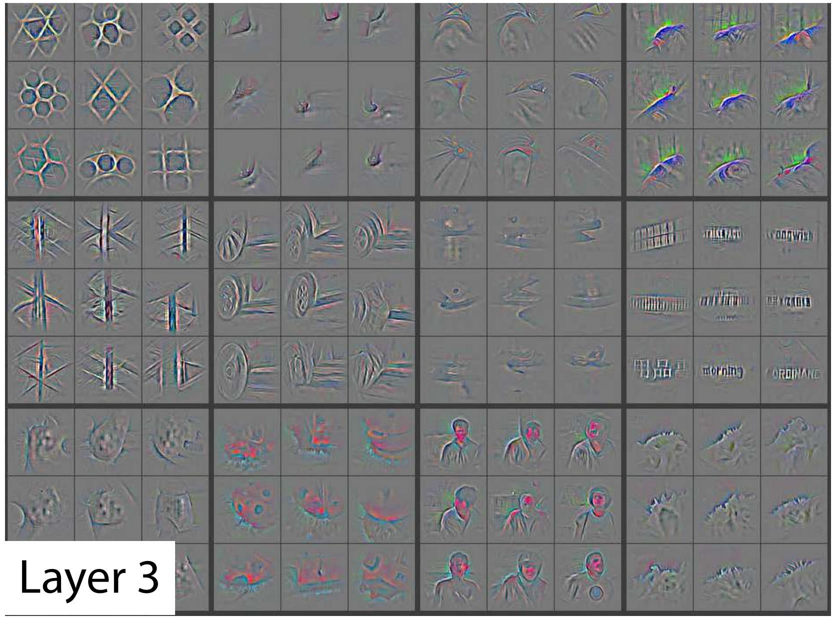
- Most of the aggregated values at neurons do not result in activations
- Find image patches in dataset that maximally activate/excite a unit

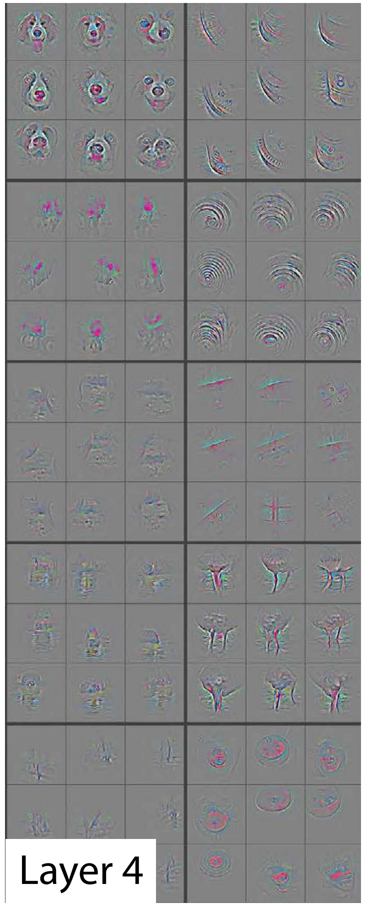


Feature extraction evolution

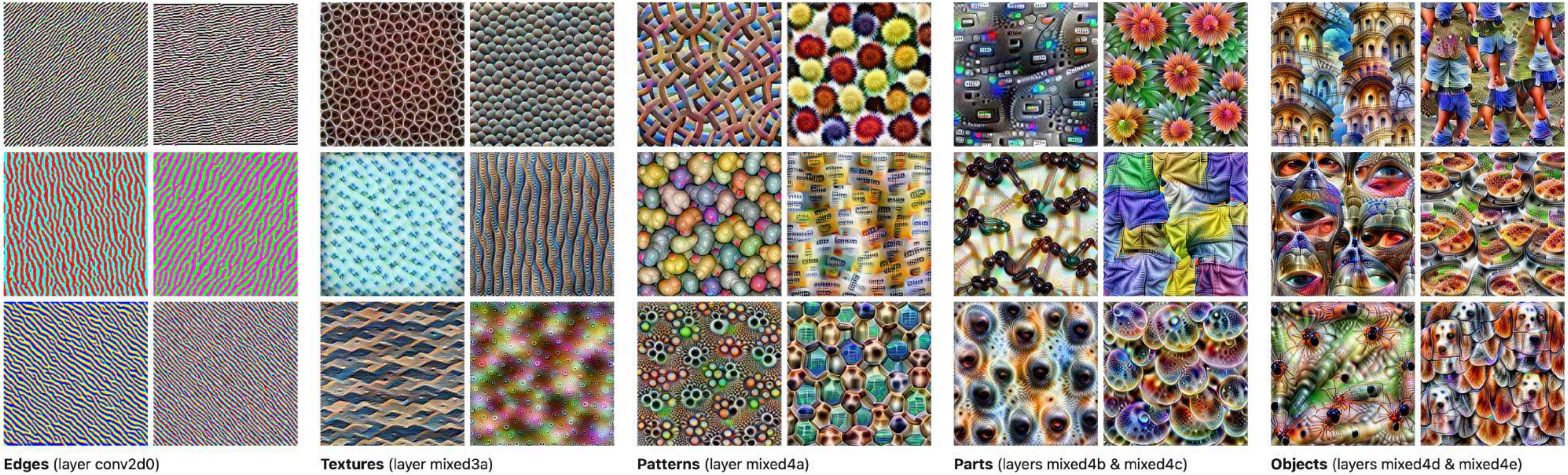
- Lower layers extract lower-level features
- Higher layers compose extracted features to compose high-level features





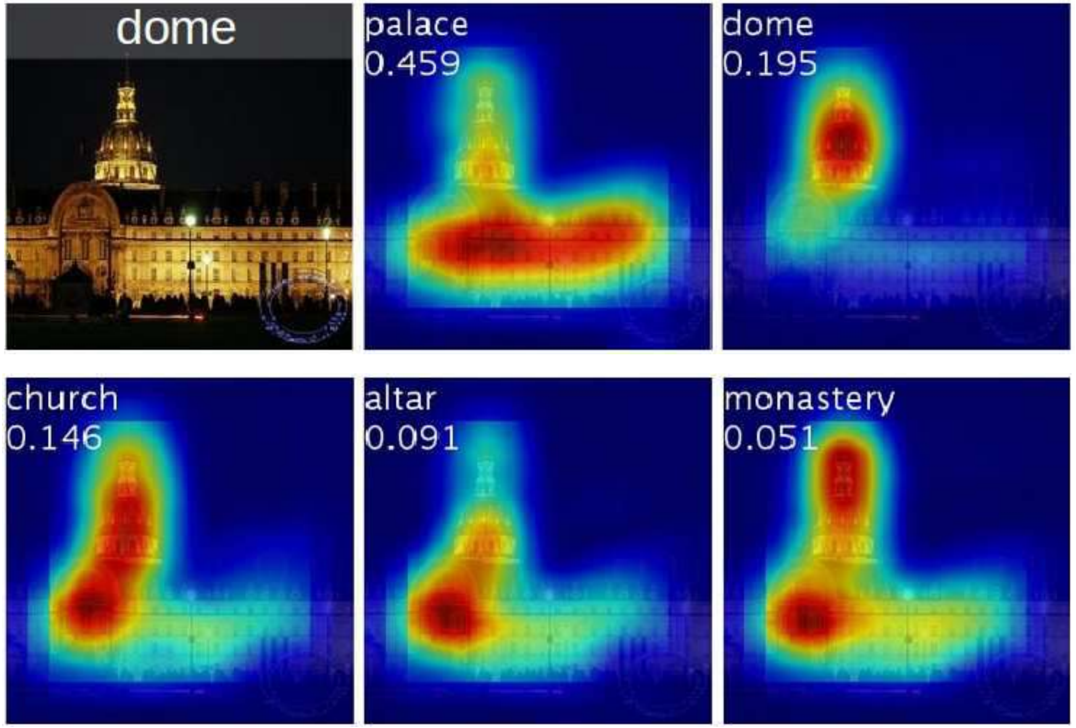


Layerwise Visualisation of CNNs



Class Activation Maps

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

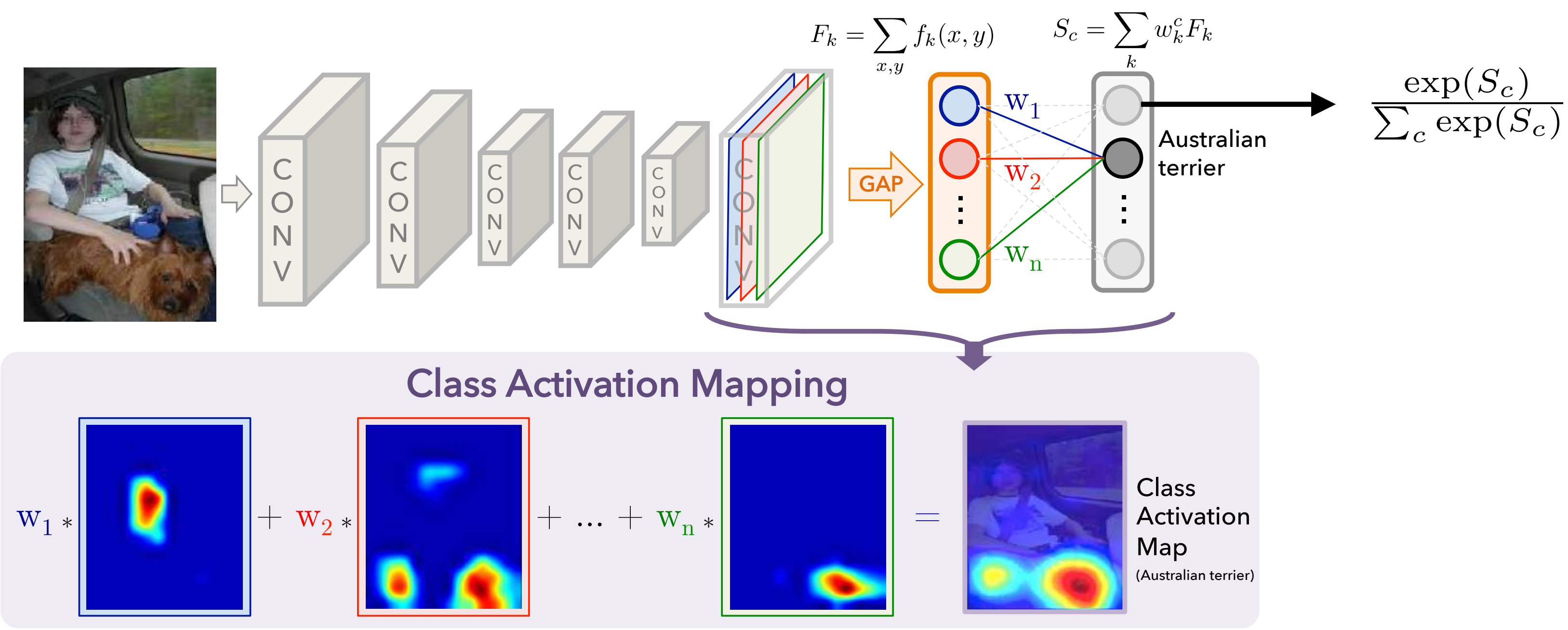


[Zhou et al. CVPR '16] https://arxiv.org/pdf/1512.04150.pdf

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

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



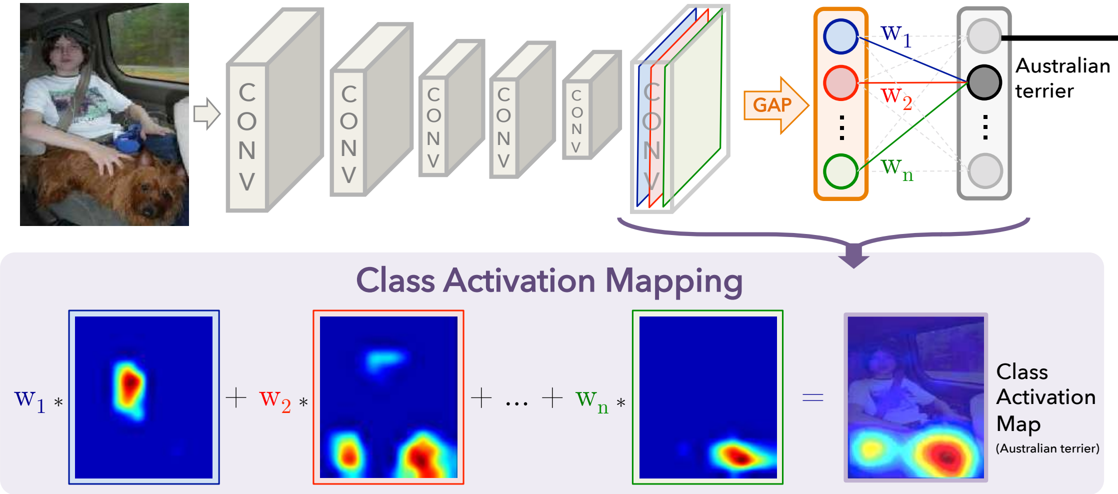
[Zhou et al. CVPR '16] https://arxiv.org/pdf/1512.04150.pdf

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

$$f_k(x,y)$$

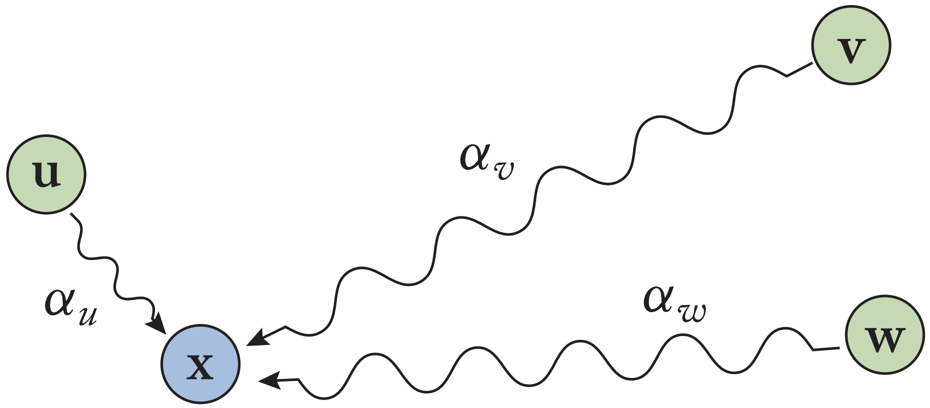
$$\sum_{x,y} f_k(x,y)$$



$$\frac{\exp(S_c)}{\sum_c \exp(S_c)}$$

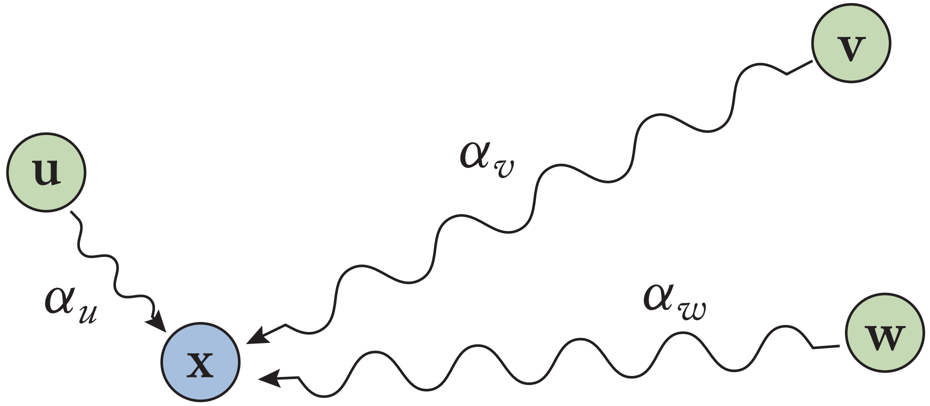
Attention in Language

- 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



New contextual representation $\mathbf{x}' = \alpha_u\,\mathbf{u} + \alpha_v\,\mathbf{v} + \alpha_w\,\mathbf{w}$

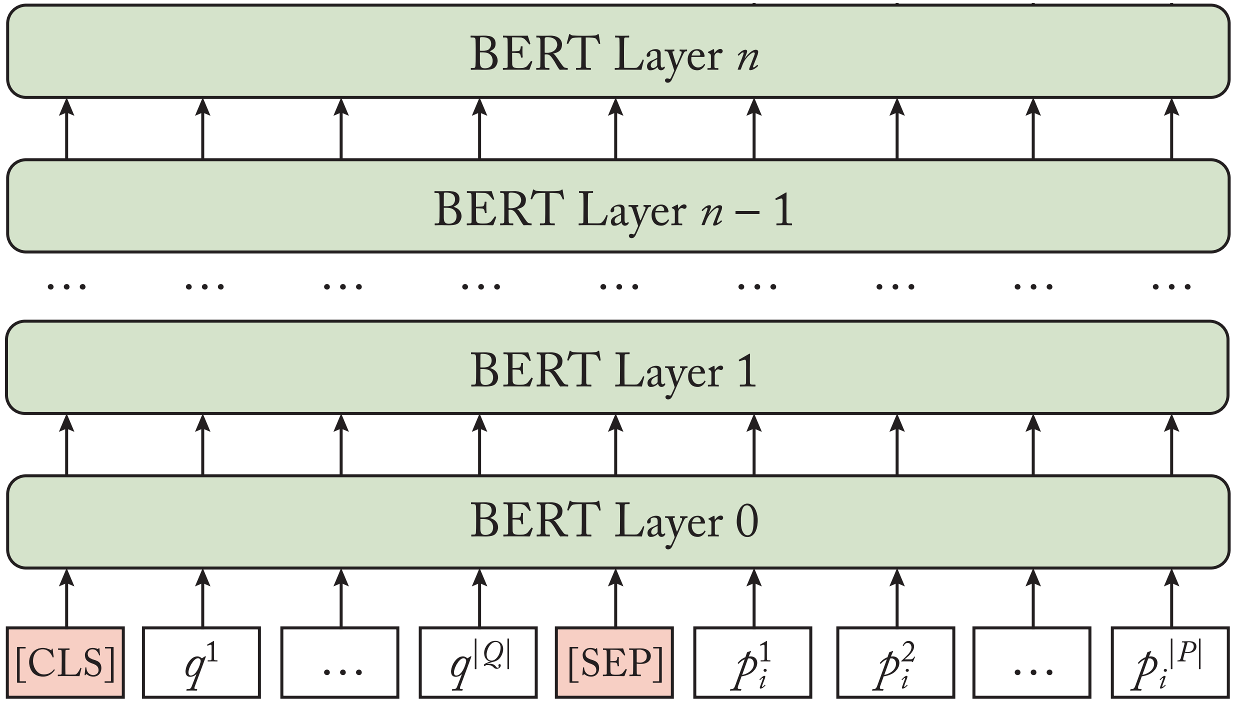
- Attention mechanism in neural language models is crucial for extracting latent features
- Self attention in language is aimed at re-representing the initial representation based on the context
- Neural models consume non-contextual token-level representations and output contextual token-level representation

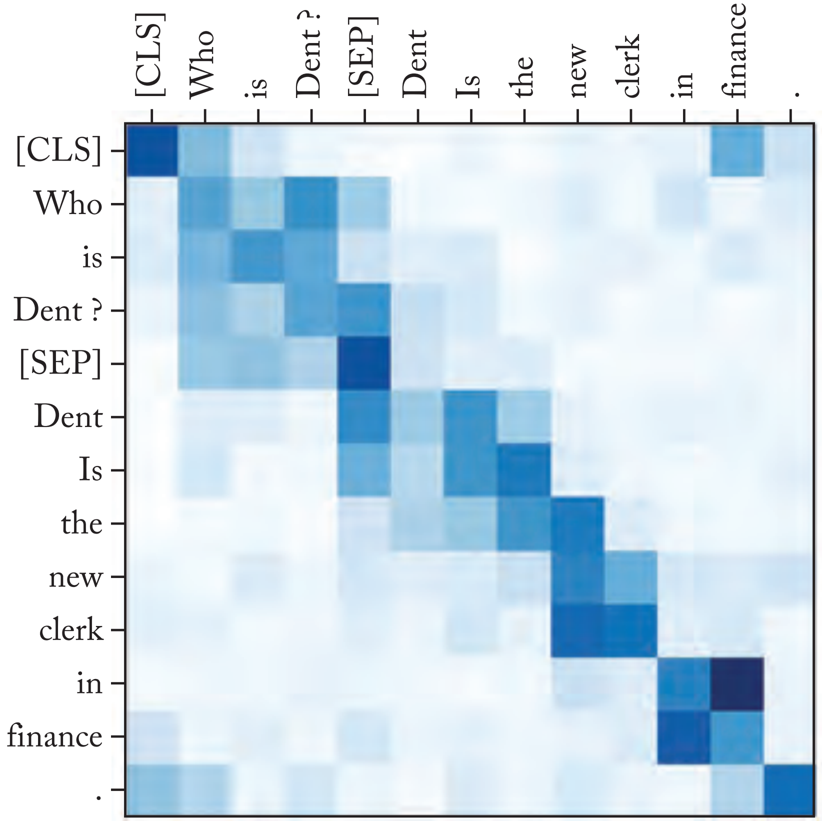


New contextual representation $\mathbf{x}' = \alpha_u \mathbf{u} + \alpha_v \mathbf{v} + \alpha_w \mathbf{w}$

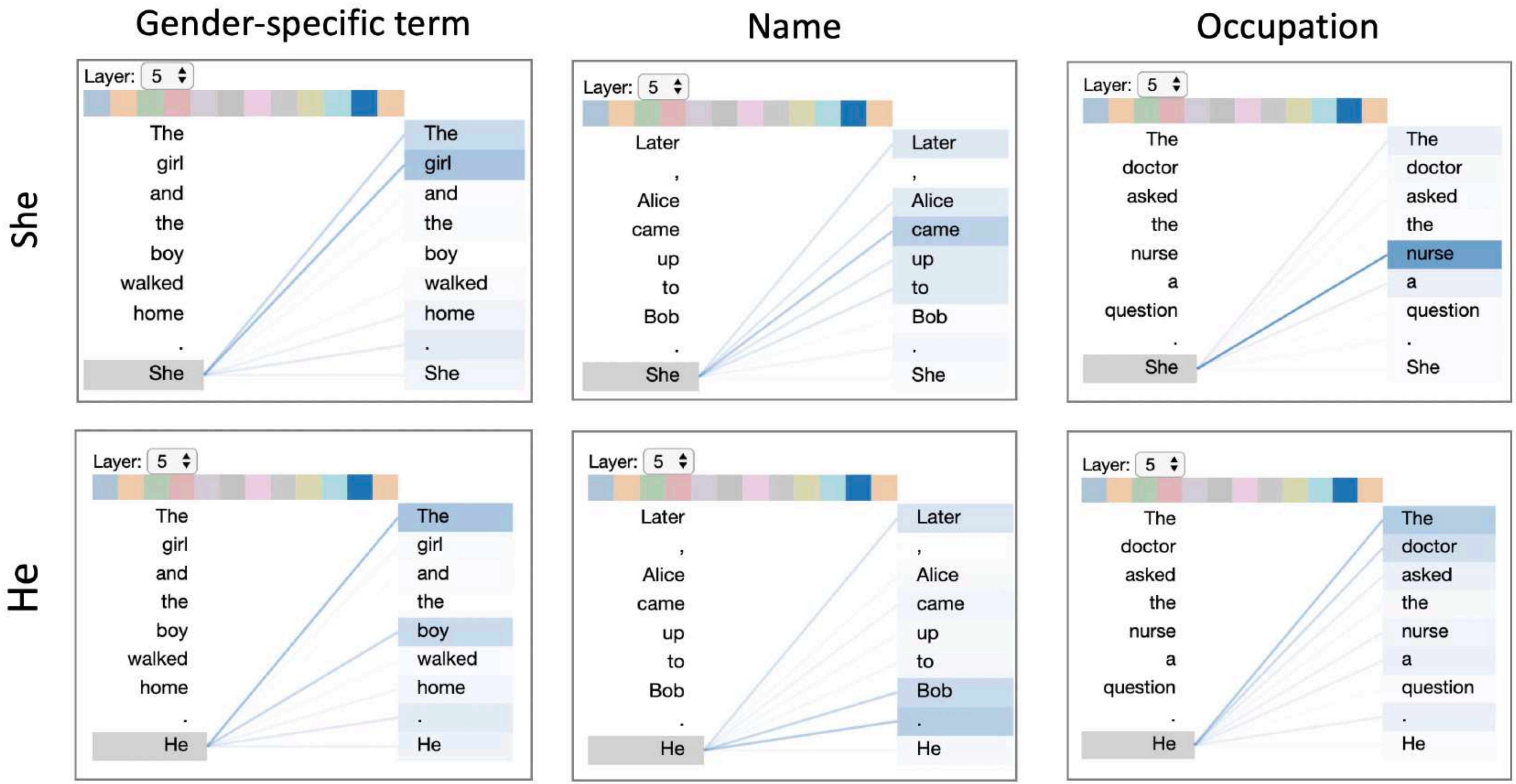
$$\alpha_u = \frac{e^{\mathrm{sim}(\mathbf{u},\mathbf{x})}}{e^{\mathrm{sim}(\mathbf{u},\mathbf{x})} + e^{\mathrm{sim}(\mathbf{v},\mathbf{x})} + e^{\mathrm{sim}(\mathbf{w},\mathbf{x})}}; \quad \mathrm{sim}(\mathbf{u},\mathbf{x}) = \mathbf{x} \cdot \mathbf{W}\mathbf{u}$$

Attention Maps in Transformers





Visualizing Attention Units



Other Interactive visualisations

- Interactive visualisation 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
- ...

iML: Post-hoc Methods for Neural Networks

Simple gradients, Integrated gradients

Marius Lindauer and Avishek Anand

Leibniz Universität Hannover

Winter Term 2021

Sensitivity Analysis

- Neural Networks are differentiable machines
  - The output can be written as a function of the parameters and input
  - One can differentiate the output function w.r.t parameters
  - The underlying idea is used for training Neural Nets using gradient descent

$$f(x;\theta) \qquad\qquad \frac{\partial f(x;\theta)}{\partial \theta}$$

- **Sensitivity Analysis:** How sensitive is the output $f()$ w.r.t to a small change in the input ?

Sensitivity Analysis

- Neural Networks are differentiable machines
  - The output can be written as a function of the parameters and input
  - One can differentiate the output function w.r.t parameters
  - The underlying idea is used for training Neural Nets using gradient descent

$$f(x;\theta) \qquad\qquad \frac{\partial f(x;\theta)}{\partial \theta}$$

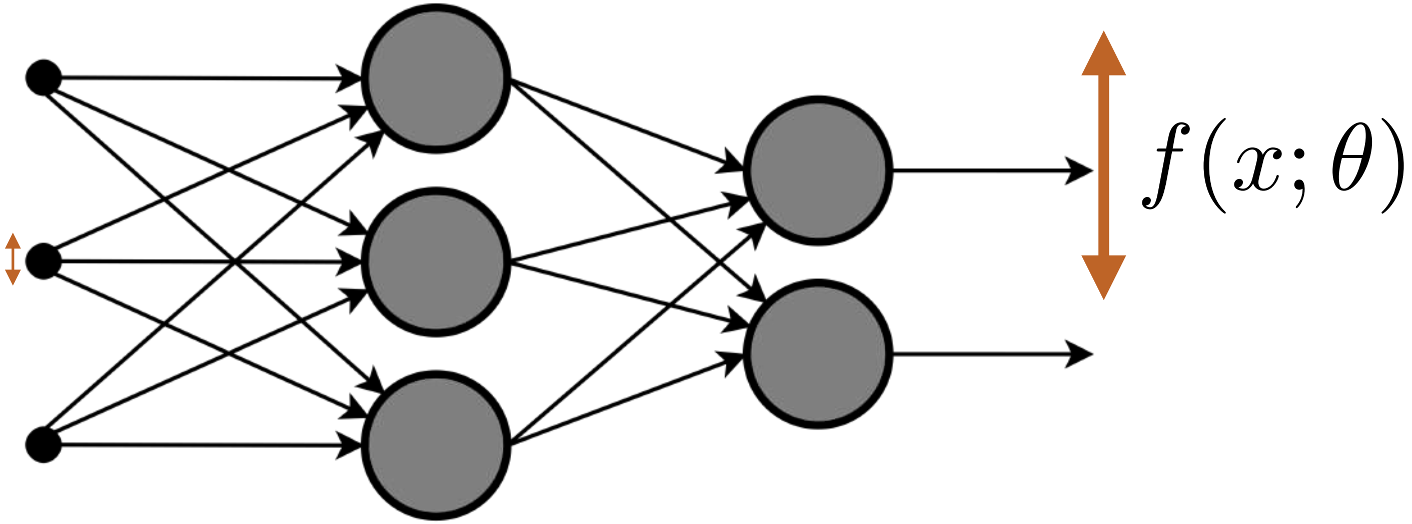
- **Sensitivity Analysis:** How sensitive is the output $f()$ w.r.t to a small change in the input ?

$$\frac{\partial f(x;\theta)}{\partial x}$$

Sensitivity Analysis

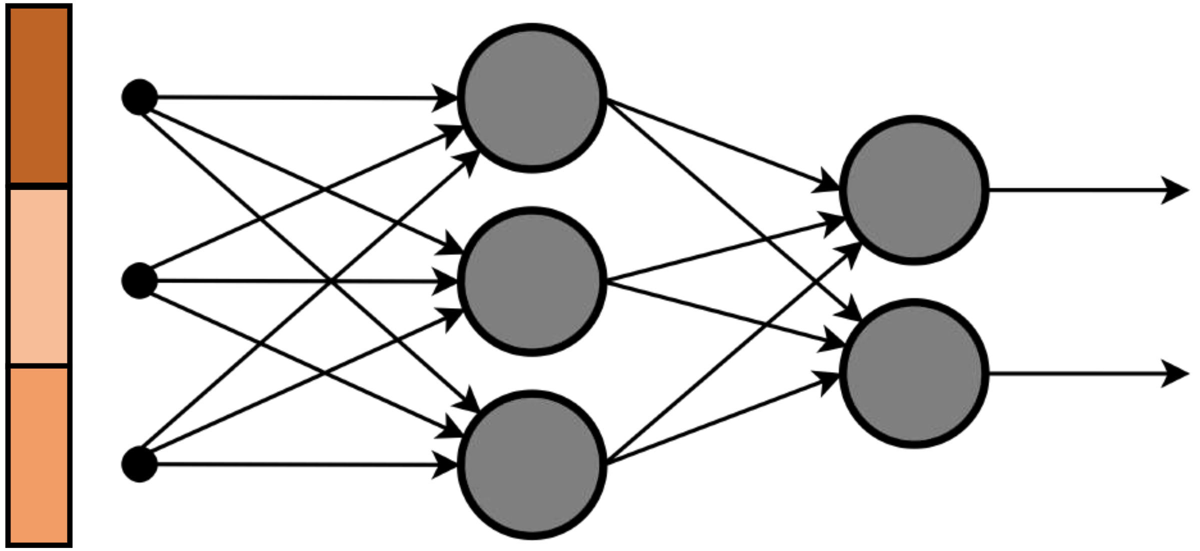
- How sensitive is the output $f()$ w.r.t to a small change in the input ?
  - If a small change in the input feature causes a large change in output, then that feature is **responsible** for the prediction
  - Back-propagation into the input: instead of computing $\frac{\partial f(x;\theta)}{\partial \theta}$

$$\frac{\partial f(x;\theta)}{\partial x}$$



Saliency Maps

- Visualize the gradients over each feature
  - as a heat map or Saliency Maps
  - Saliency maps are feature attribution methods that are based on gradients



$$f(x;\theta)$$

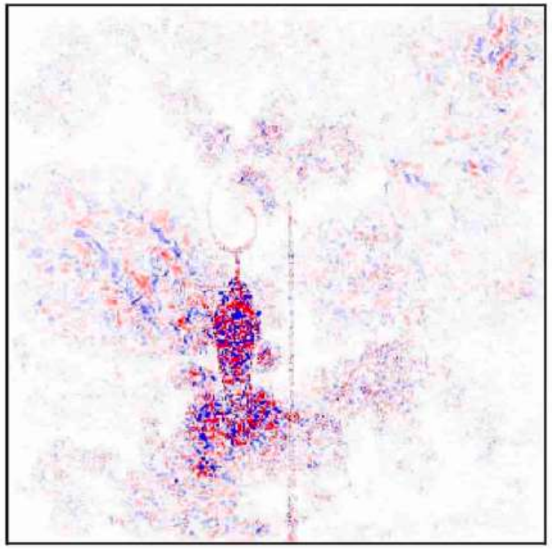
[Simonyan, Vedaldi, Zisserman '14]

Saliency Maps for Images

- Images have multiple channels where each channel is a 2-D matrix

$$M_{ij} = \max_c |\nabla_x S_c(X)|_{(i,j,c)}$$

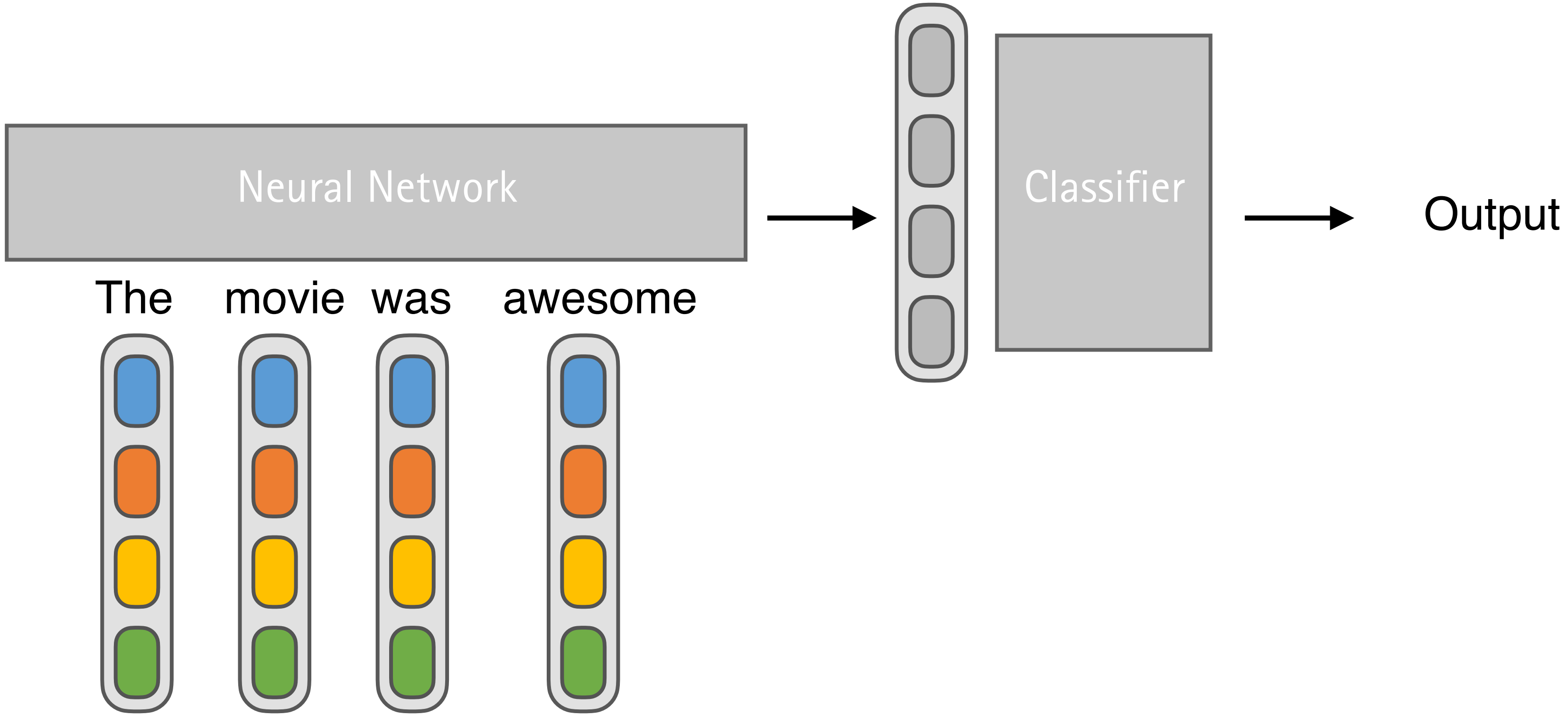
$$M_{ij} = \max_c |\nabla_x S_c(X)|_{(i,j,c)}$$



[Simonyan, Vedaldi, Zisserman '14]

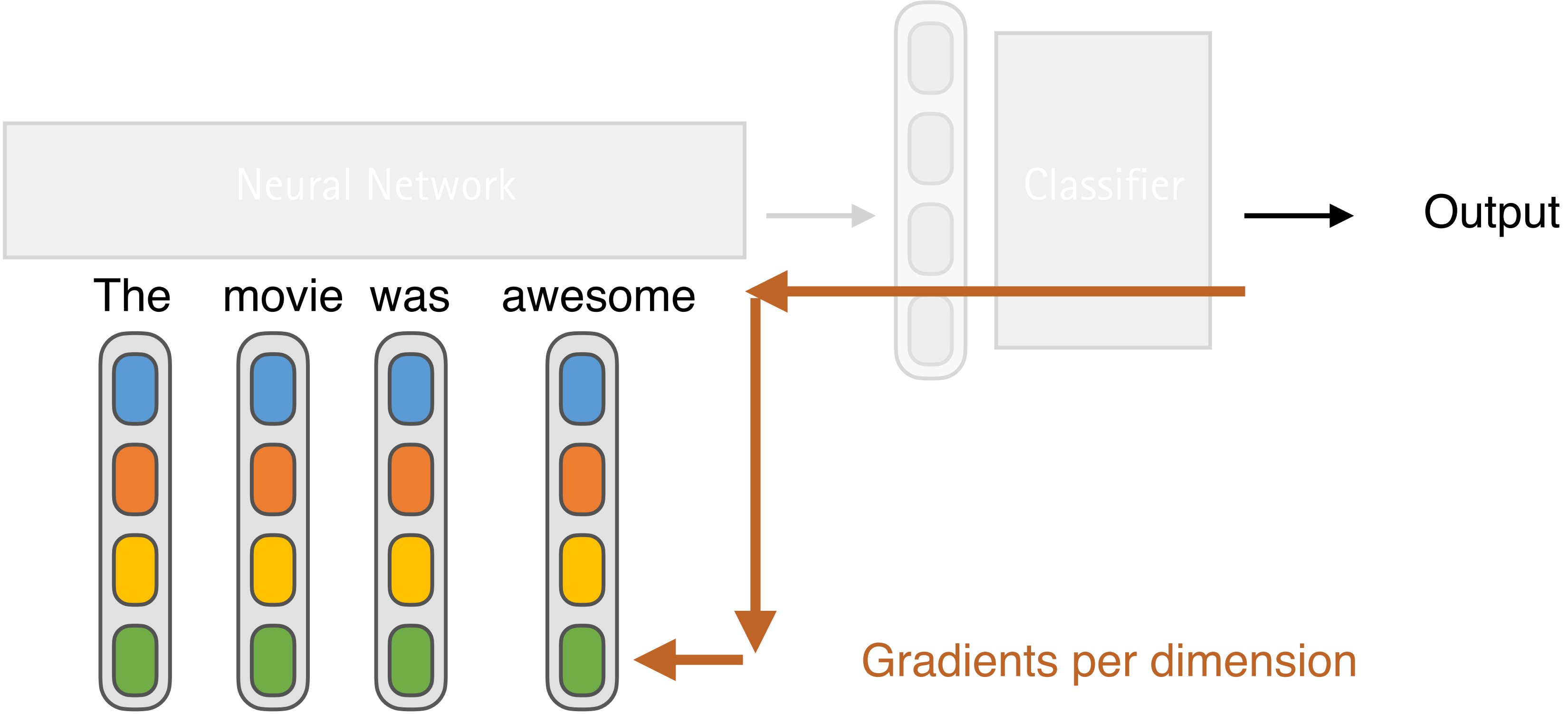
Saliency Maps for Language

- Words are associated with an embedding
- Computing gradients back to the inputs is different in comparison to images



Saliency Maps for Language

- We obtain gradients per dimension but we want attributions or importance scores at the level of world
- **Idea:** Simple aggregations of dimension-level gradients like sum, average, etc.



[Simonyan, Vedaldi, Zisserman '14]

Saliency Maps – Setting

Which features are responsible for the decision given..

A trained model **M**	Post-hoc interpretability
An instance **x**	Local interpretability
Access to model parameters	White-box interpretability

Saliency Maps – Setting

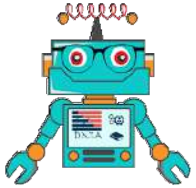
Which features are responsible for the decision given..

A trained model M — Post-hoc interpretability

An instance x — Local interpretability

Access to model parameters — White-box interpretability

Input



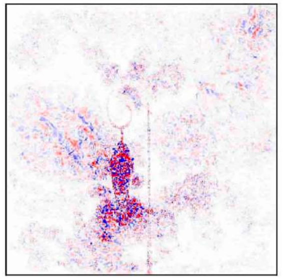
"Pole"

Output

Saliency Maps

Heatmaps

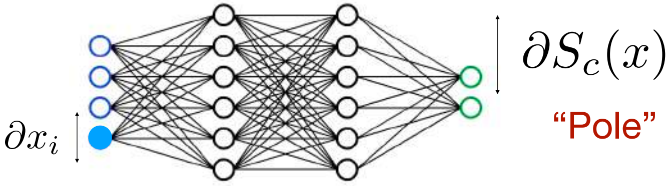
Feature Attributions



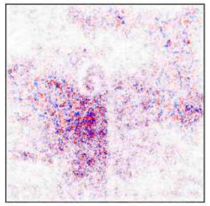
Saliency Maps – Setting

Which features are responsible for the decision given..

A trained model **S** An instance **x** Access to model parameters



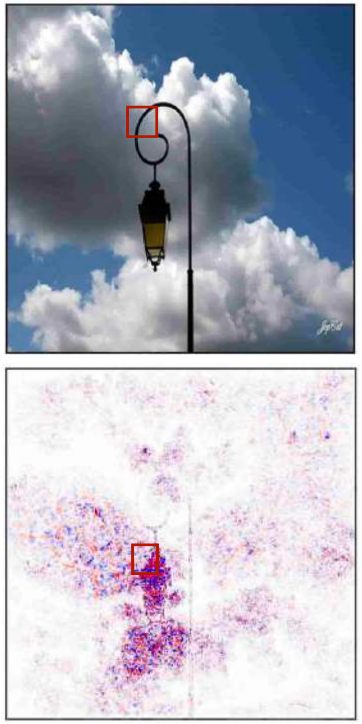
A feature is more relevant if a small perturbation causes large change in the output



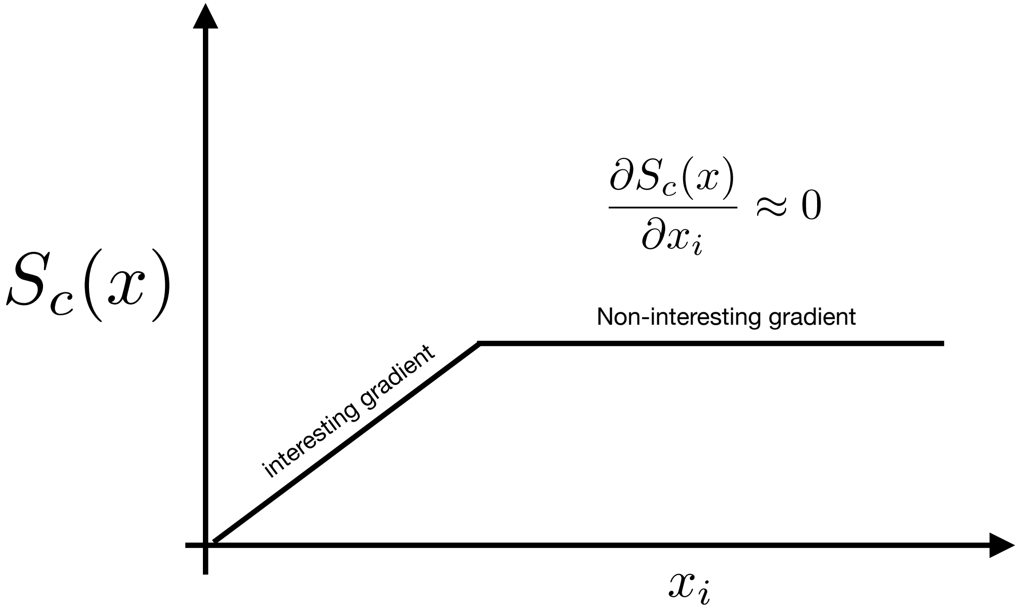
Saliency Map

$$R_i^c(x) = \frac{\partial S_c(x)}{\partial x_i}$$

Problems with Deep Nets



$$R_i^c(x) = \frac{\partial S_c(x)}{\partial x_i}$$

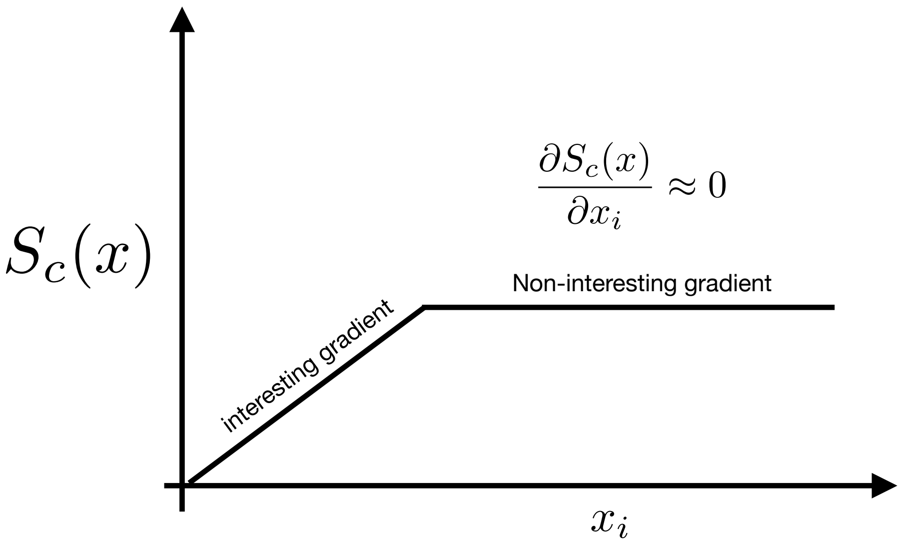


Deep Neural Networks are usually trained till "Saturation"

Perturbing Inputs

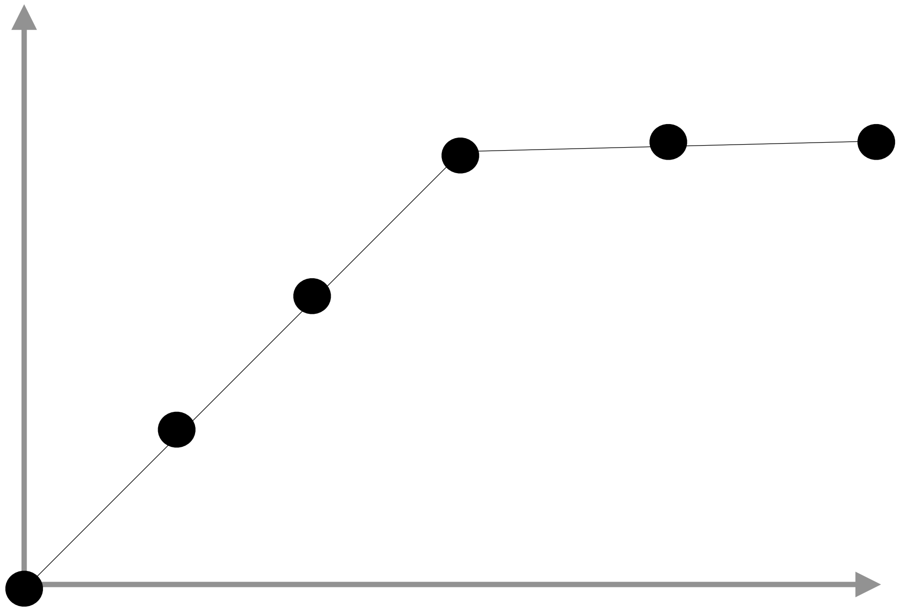
- **Small perturbations** at the saturation point do not give us interesting gradients
- Extreme perturbation (to say a baseline image) can give us interesting gradients

$$R_i^c(x) = \frac{\partial S_c(x)}{\partial x_i}$$



Integrated Gradients

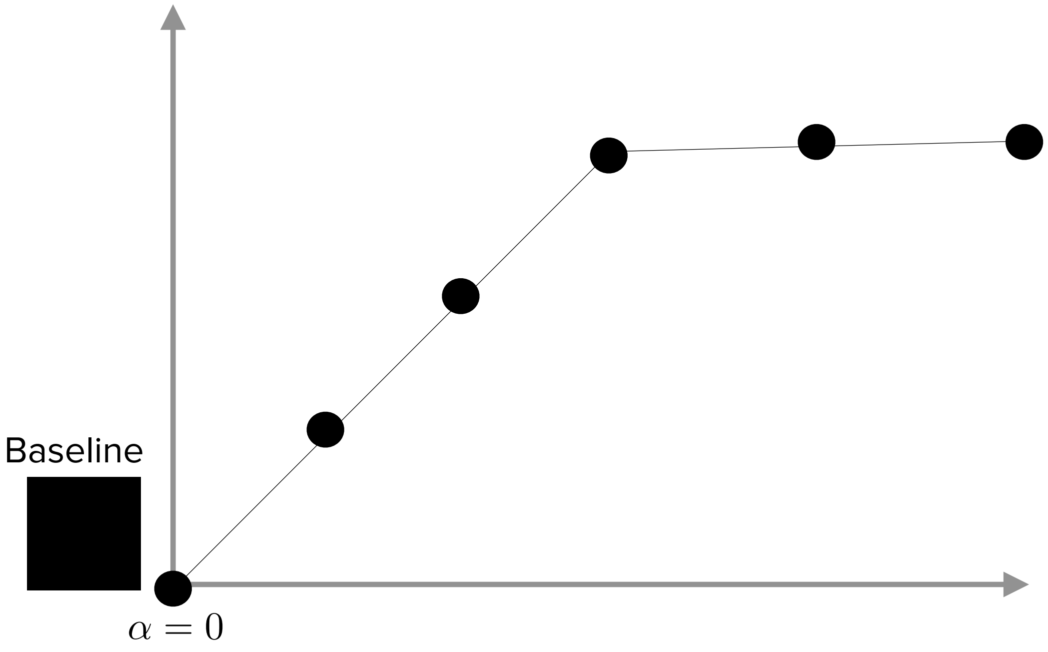
Compute gradient estimate based on gradients over a path of specific perturbations



Integrated Gradients

Compute gradient estimate based on gradients over a path of specific perturbations

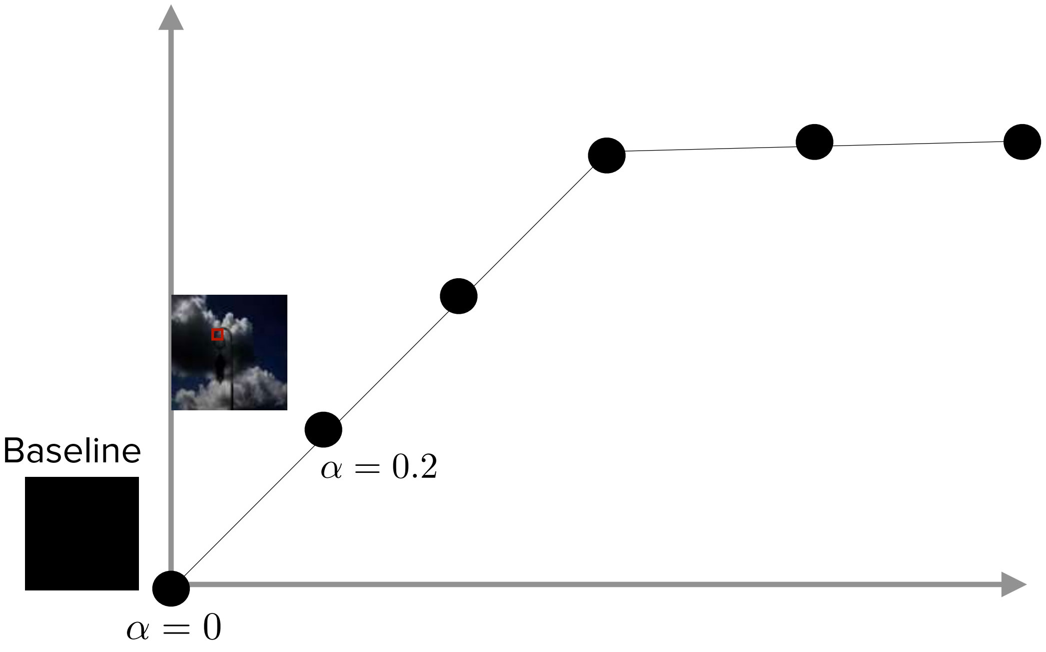
Choose a Baseline to contrast



Integrated Gradients

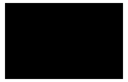
Compute gradient estimate based on gradients over a path of specific perturbations

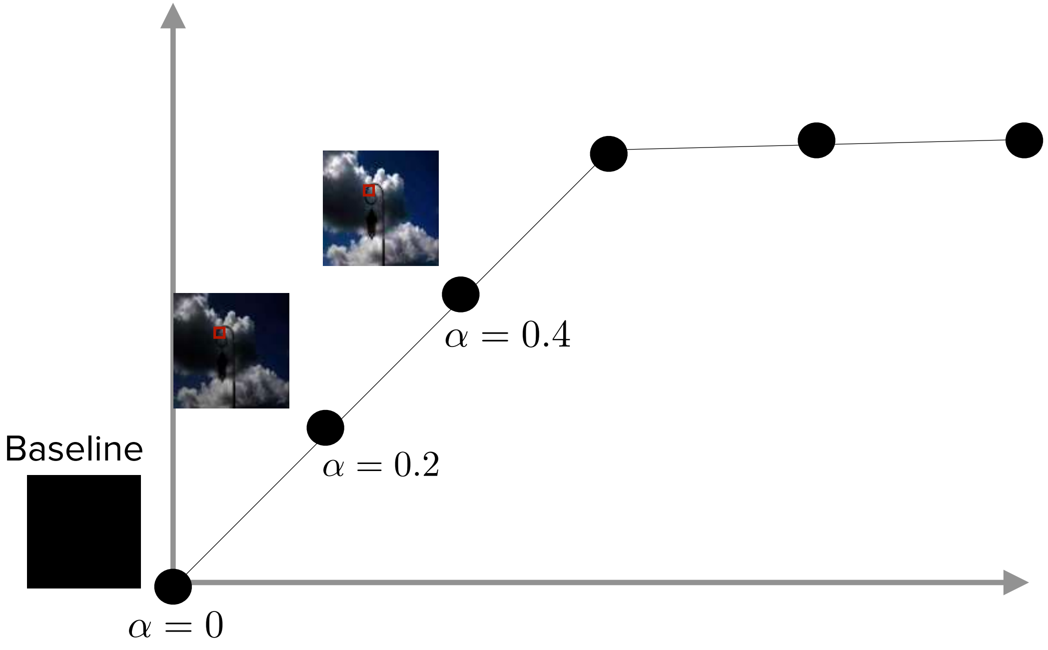
Choose a Baseline to contrast



Integrated Gradients

Compute gradient estimate based on gradients over a path of specific perturbations

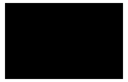
Choose a Baseline to contrast 

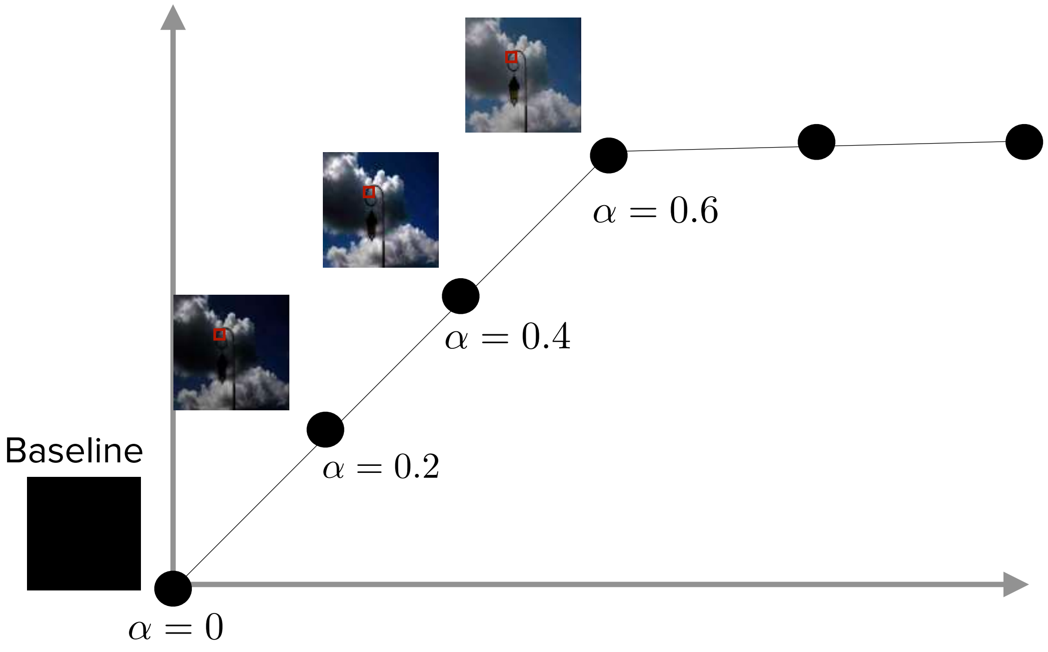


[Sundarajan et al. ’16]

Integrated Gradients

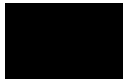
Compute gradient estimate based on gradients over a path of specific perturbations

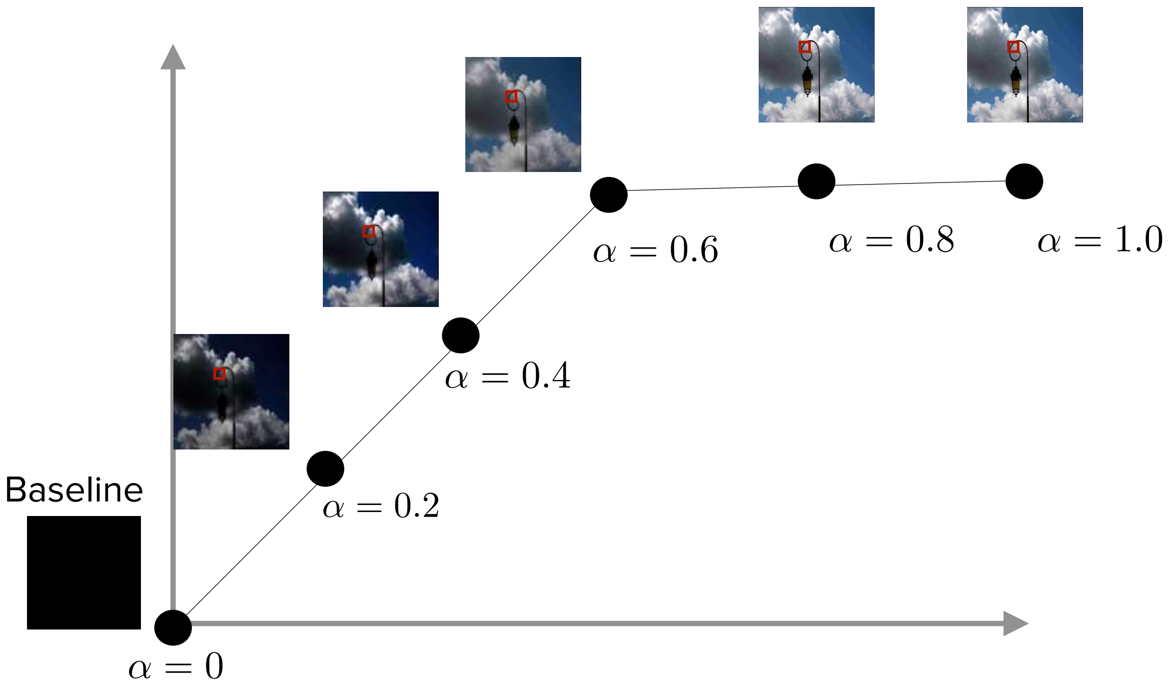
Choose a Baseline to contrast 



Integrated Gradients

Compute gradient estimate based on gradients over a path of specific perturbations

Choose a Baseline to contrast 



Integrated Gradients

Leibniz Universität Hannover

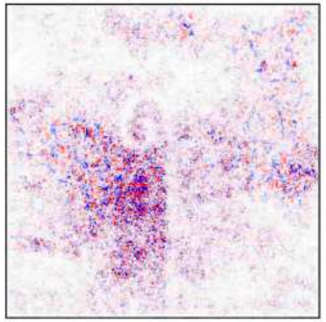
1. Choose a Baseline to contrast
2. Compute gradients at different mask values
3. Attribution = Aggregation over gradients computed for a certain set of perturbations

$$R_i^c(x) = x_i \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial\left(\tilde{x}_i\right)} \bigg|_{\tilde{x}=\underbrace{\bar{x}}_{\text{Baseline}}+\alpha(\underbrace{x}_{\text{Original}}-\bar{x})} d\alpha$$

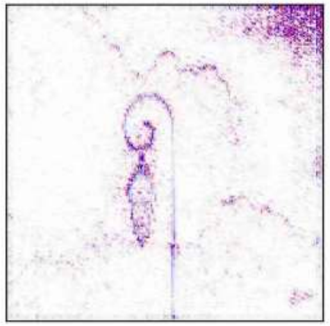
Integrated Gradients monitors how the network changes from a zero signal input to actual input through the use of gradients

Baseline

- Baseline is an information less input
- The choice of baselines matters a lot and is typically domain dependent
    - Black or gray images
    - Zero embedding in language
    - Random document in retrieval



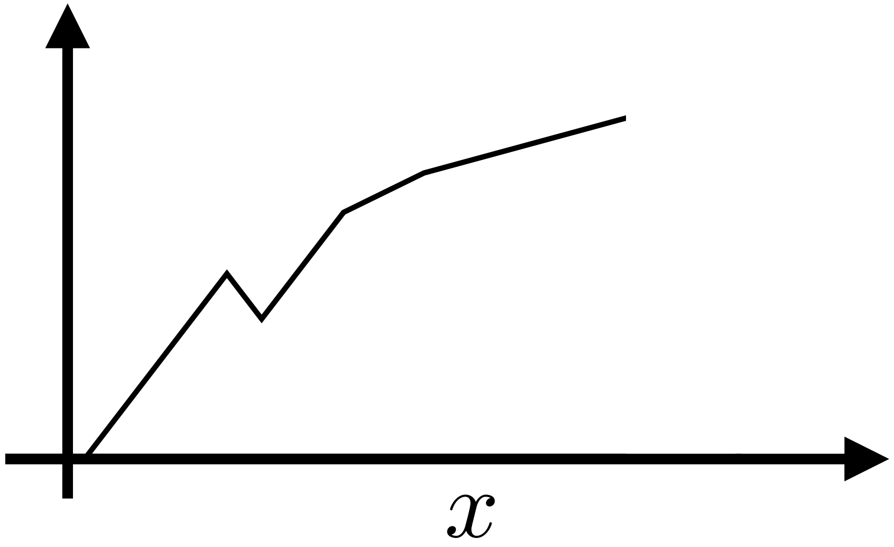
Simple Gradient



Integrated Gradients

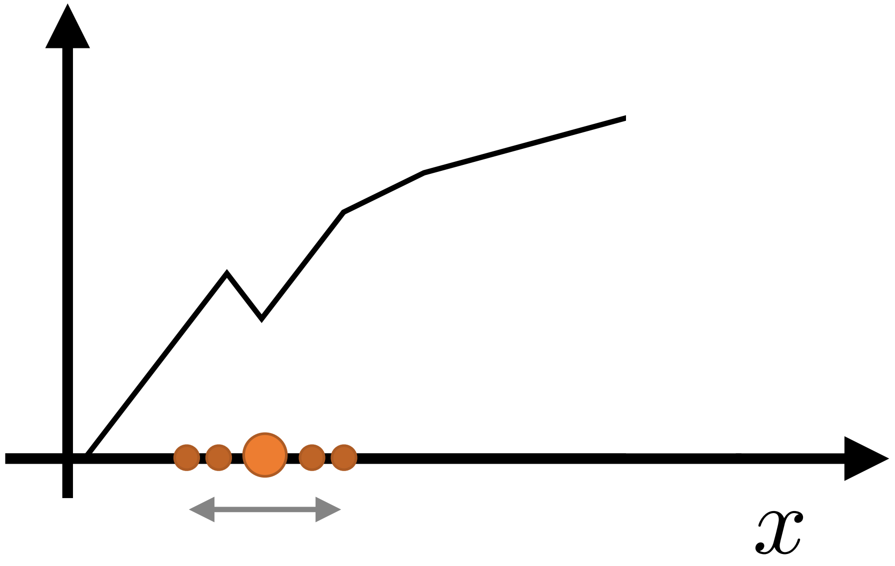
SmoothGrad

- Gradients are local ways to measure sensitivity
- In highly nonlinear loss surfaces you obtain quite noisy gradients
  - In this figure, majority of the neighbourhood gives positive gradient



SmoothGrad

- Calculate multiple copies of the input with a small noise (usually gaussian noise)
- Actual gradient is the average of the gradients of each of the copies



Conclusion

- Gradients are central in computing feature attributions and are visualised using saliency maps
- Simple gradient-based approaches for neural networks attribute the importance back to the input features
- Deep learning models suffer from critical problems for gradient-based methods
  - Models are trained to saturation given near-zero gradients – Integrated Gradients
  - Gradients are unstable due to highly non-linear loss surface – SmoothGrad
- Tons of other approaches proposed in the literature
- Caution that explanations might disagree with each other
- Caution that gradient-based approaches need to be adapted depending on the input style