ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

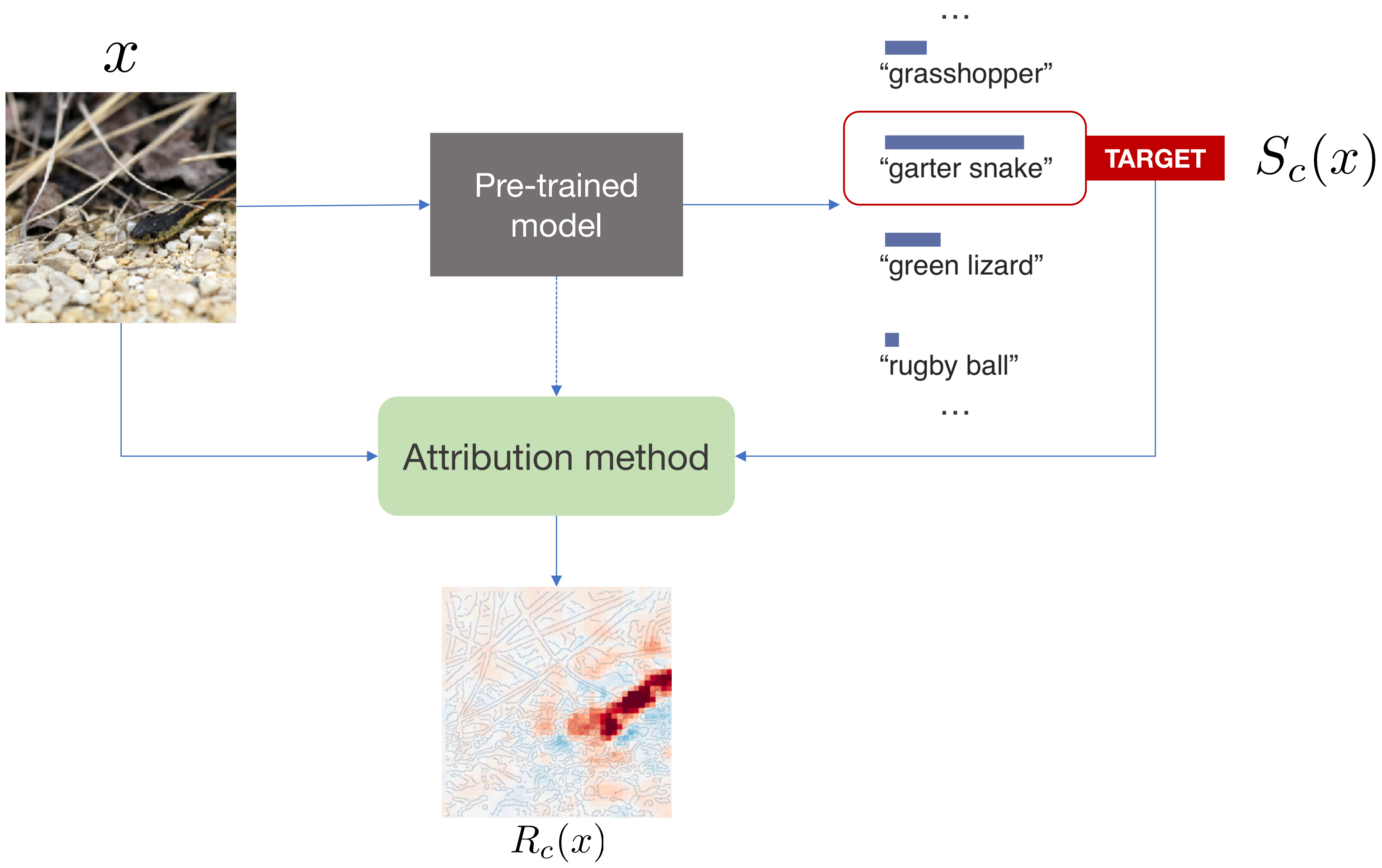
Disney RESEARCH
STUDIOS

Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Values Approximation

Marco Ancona, Cengiz Öztireli[2], Markus Gross[1,2]

[1]Department of Computer Science, ETH Zurich, Switzerland

[2]Disney Research, Zurich, Switzerland


x
Pre-trained model
...
"grasshopper"
"garter snake"
TARGET
$S_c(x)$
"green lizard"
"rugby ball"
...
Attribution method
$R_c(x)$

Attribution methods

Saliency Maps
Simonyan et al. 2015

Integrated Gradients
Sundararajan et al. 2017

DeepLIFT
Shrikumar et al. 2017

LIME
Ribeiro et al. 2016

Gradient * Input
Shrikumar et al. 2016

Layer-wise Relevance Propagation (LRP)
Bach et al. 2015

Guided Backpropagation
Springenberg et al. 2014

Grad-CAM
Selvaraju et al. 2016

Simple occlusion
Zeiler et al. 2014

Meaningful Perturbation
Fong et al. 2017

Prediction Difference Analysis
Zintgraf et al. 2017

KernelSHAP/DeepSHAP
Lundberg et al., 2017

...

Evaluating attribution methods

- No ground-truth explanation → not easy to evaluate **empirically**

Evaluating attribution methods

- No ground-truth explanation → not easy to evaluate **empirically**
- Often based on heuristics → not easy to justify **theoretically**

Evaluating attribution methods

- No ground-truth explanation → not easy to evaluate **empirically**
- Often based on heuristics → not easy to justify **theoretically**

"Axiomatic approach"

From a set of **desired properties** to the method definition

(Some) desirable properties

Completeness

Attributions should sum up to the output of the function being considered, for comprehensive accounting.

Symmetry

If two features have exactly the same role in the model, they should receive the same attribution.

Linearity

Attributions generated for a linear combination of two models should also be a linear combination of the original attributions.

Continuity

Attributions for two nearly identical inputs on a continuous function should be nearly identical.

Shapley Values

Shapley, Lloyd S., 1953

The **only** attribution method that satisfies all the aforementioned properties.

Shapley Values

Shapley, Lloyd S., 1953

The **only** attribution method that satisfies all the aforementioned properties.

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

The function to analyze
(eg. the map from the input layer to a specific output neuron in a DNN)

Shapley Values

Shapley, Lloyd S., 1953

S is a given set of input features

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} \cdot [f(S \cup \{i\}) - f(S)]$$

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \underbrace{\sum_{S \subseteq P \setminus \{i\}}}_{} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

All unique subsets S
of features taken from the input (set) P

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \underbrace{\sum_{S \subseteq P \setminus \{i\}}}\frac{|S|!(|P|-|S|-1)!}{|P|!} \cdot \underline{[f(S \cup \{i\}) - f(S)]}$$

All unique subsets S
of features taken from the input (set) P

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

average

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

average **marginal contribution**

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

all subsets **average** **marginal contribution**

Shapley Values

Shapley, Lloyd S., 1953

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

all subsets **average** **marginal contribution**

"The **average marginal contribution** of a feature with respect to **all subsets** of other features"

Shapley Values

Shapley, Lloyd S., 1953

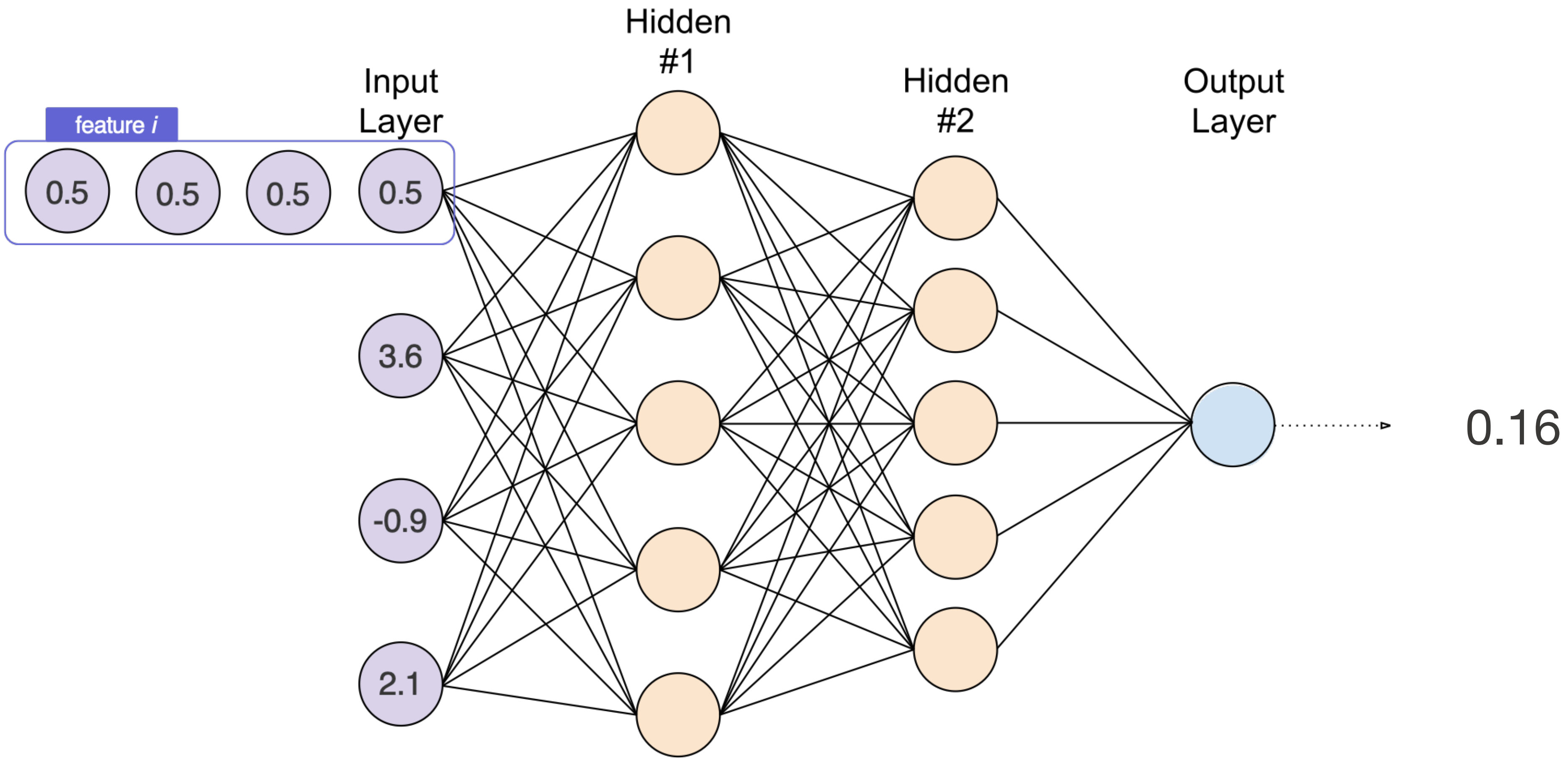
$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

Issue: testing all subsets is unfeasible!

☹

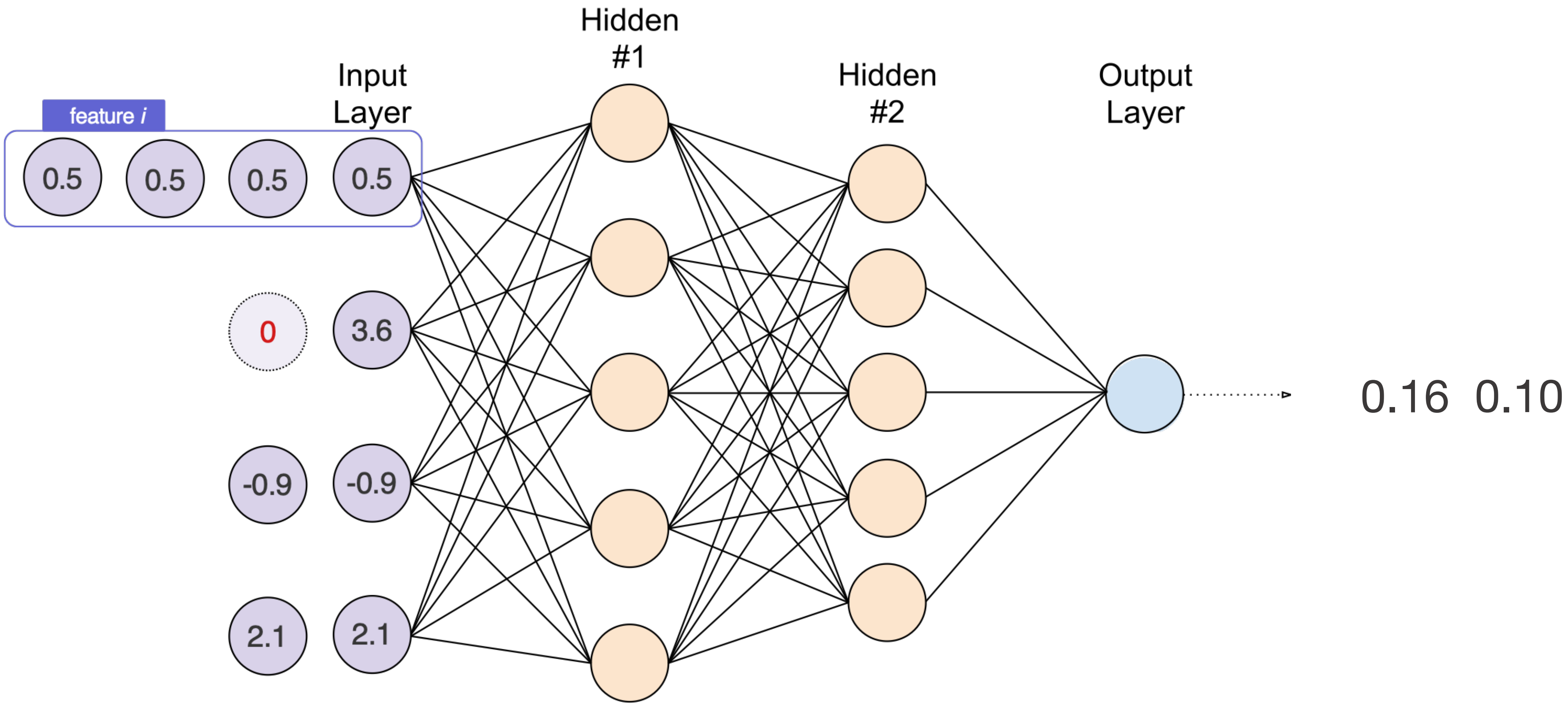
Shapley value sampling

Castro et al., 2009



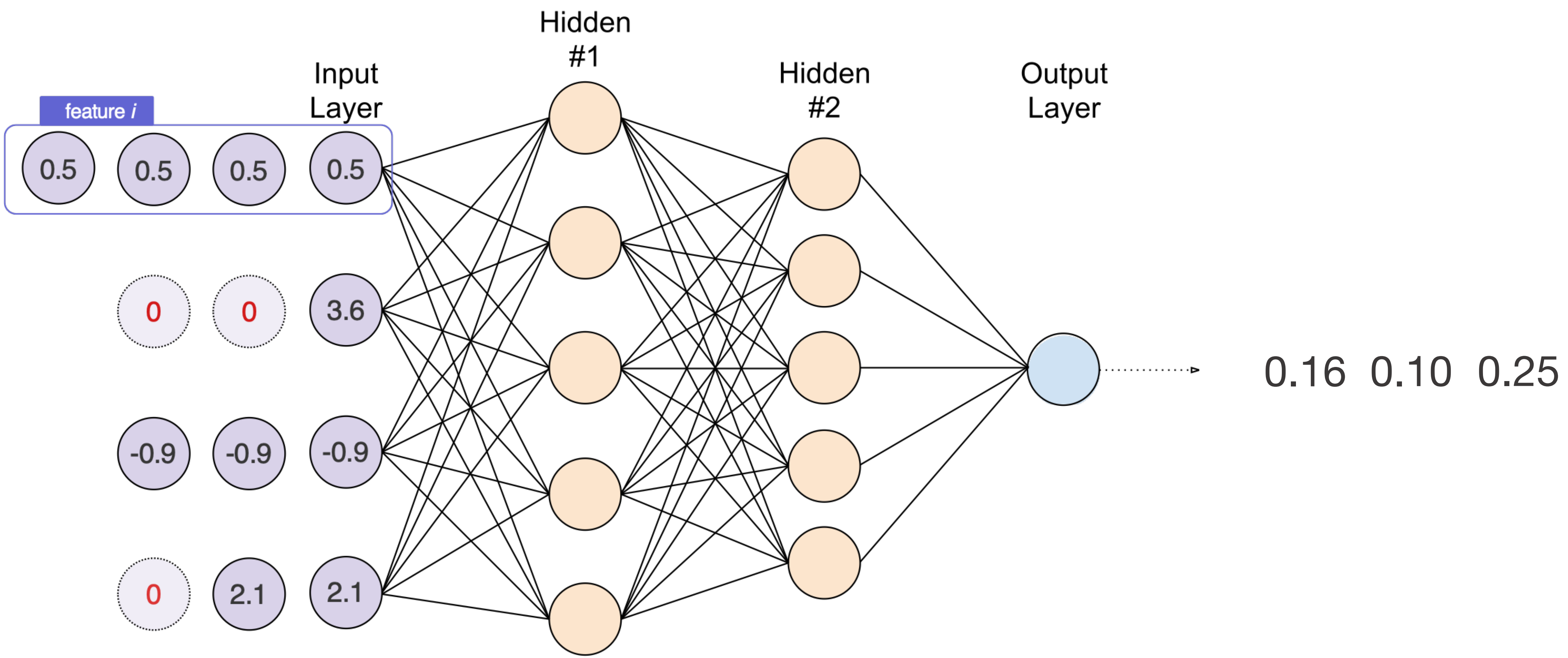
Shapley value sampling

Castro et al., 2009



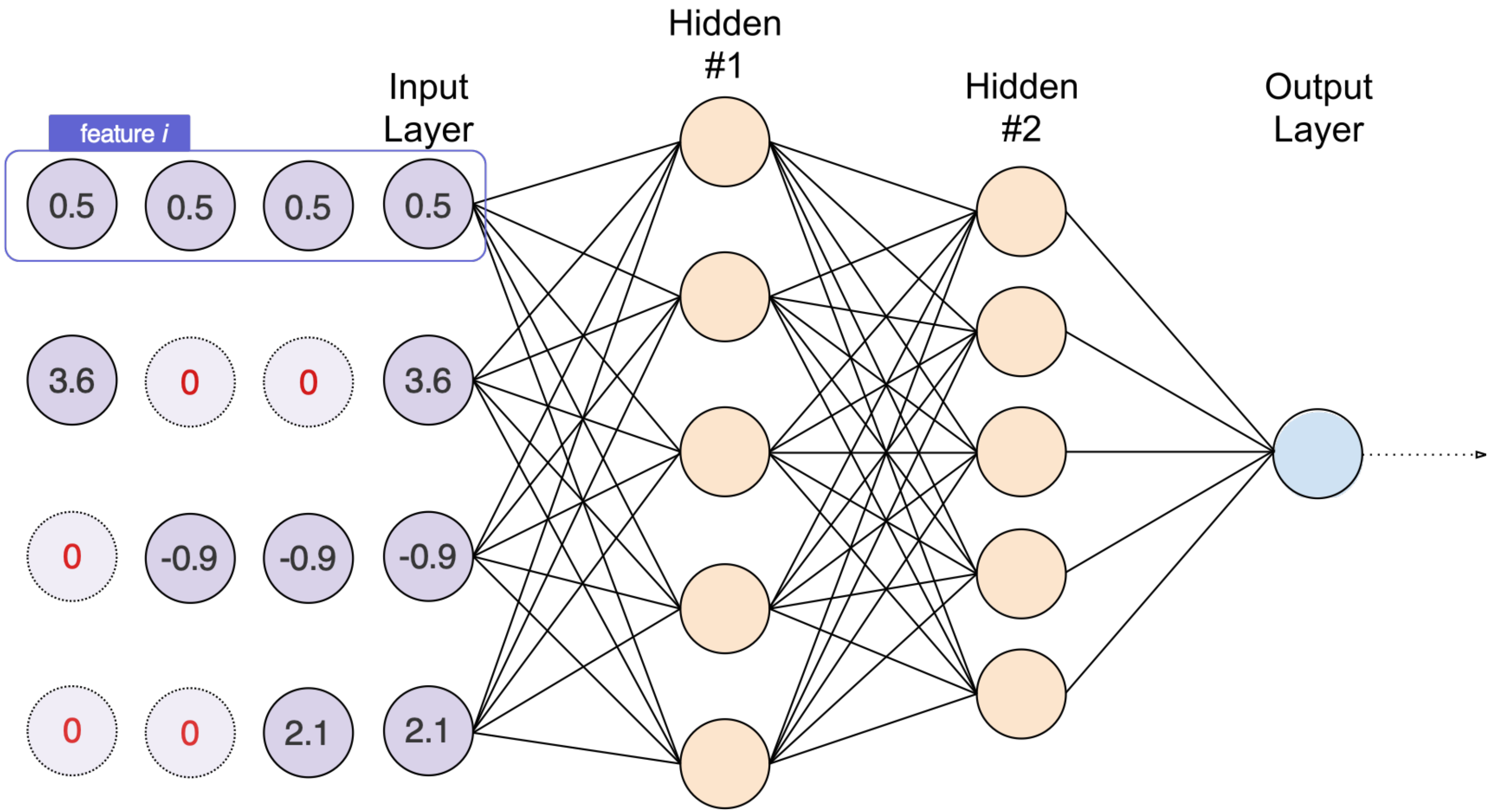
Shapley value sampling

Castro et al., 2009



Shapley value sampling

Castro et al., 2009



0.16 0.10 0.25 -0.35

Pros: Shapley value sampling is unbiased

Pros: Shapley value sampling is unbiased

Cons: might require a lot of samples (network evaluations) to produce an accurate result ☹

Pros: Shapley value sampling is unbiased

Cons: might require a lot of samples (network evaluations) to produce an accurate result ☹

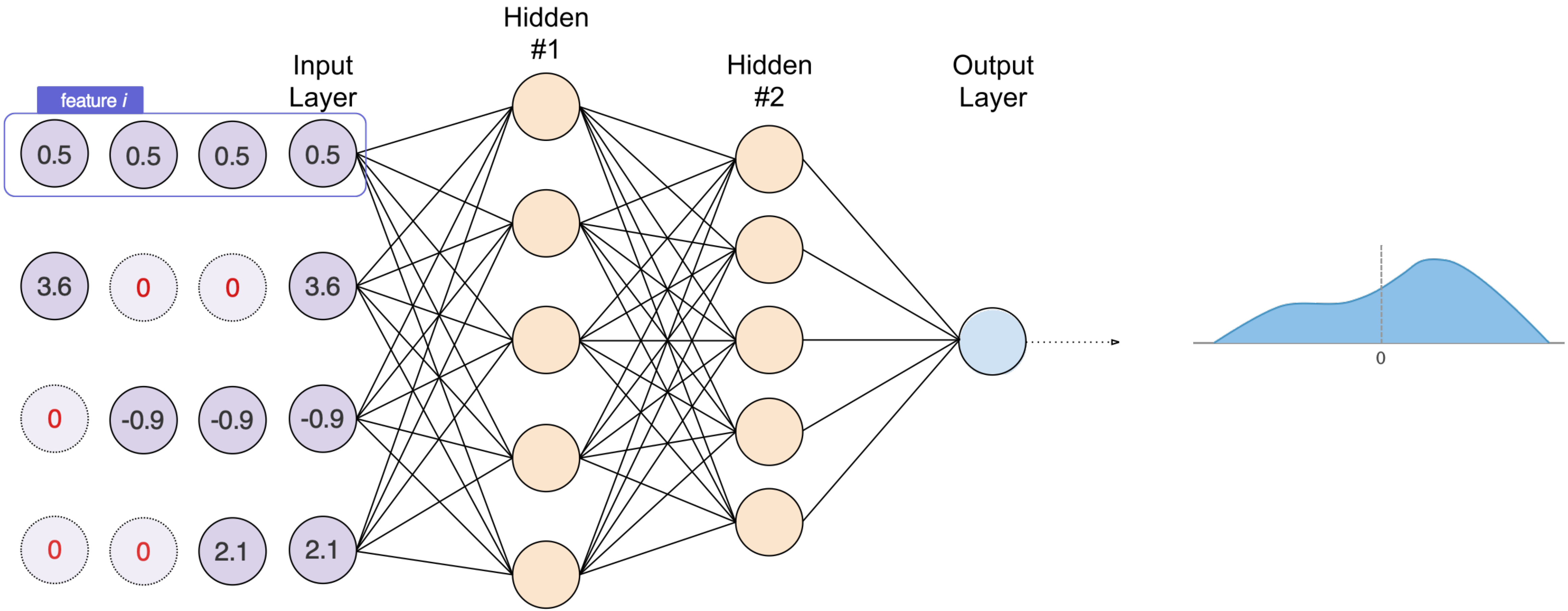
Can we avoid sampling?

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

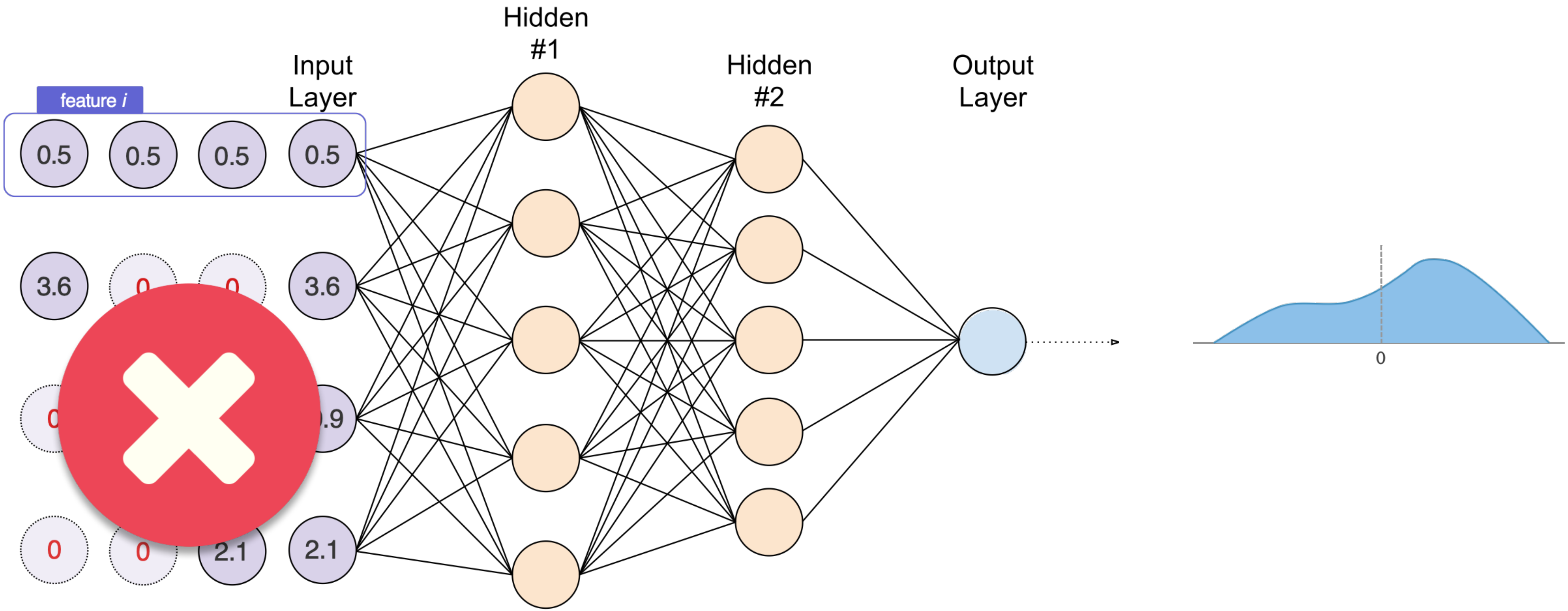
$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

$$= \mathop{\mathbb{E}}_{S \subseteq P \setminus \{i\}} f(S \cup \{i\}) - \mathop{\mathbb{E}}_{S \subseteq P \setminus \{i\}} f(S)$$

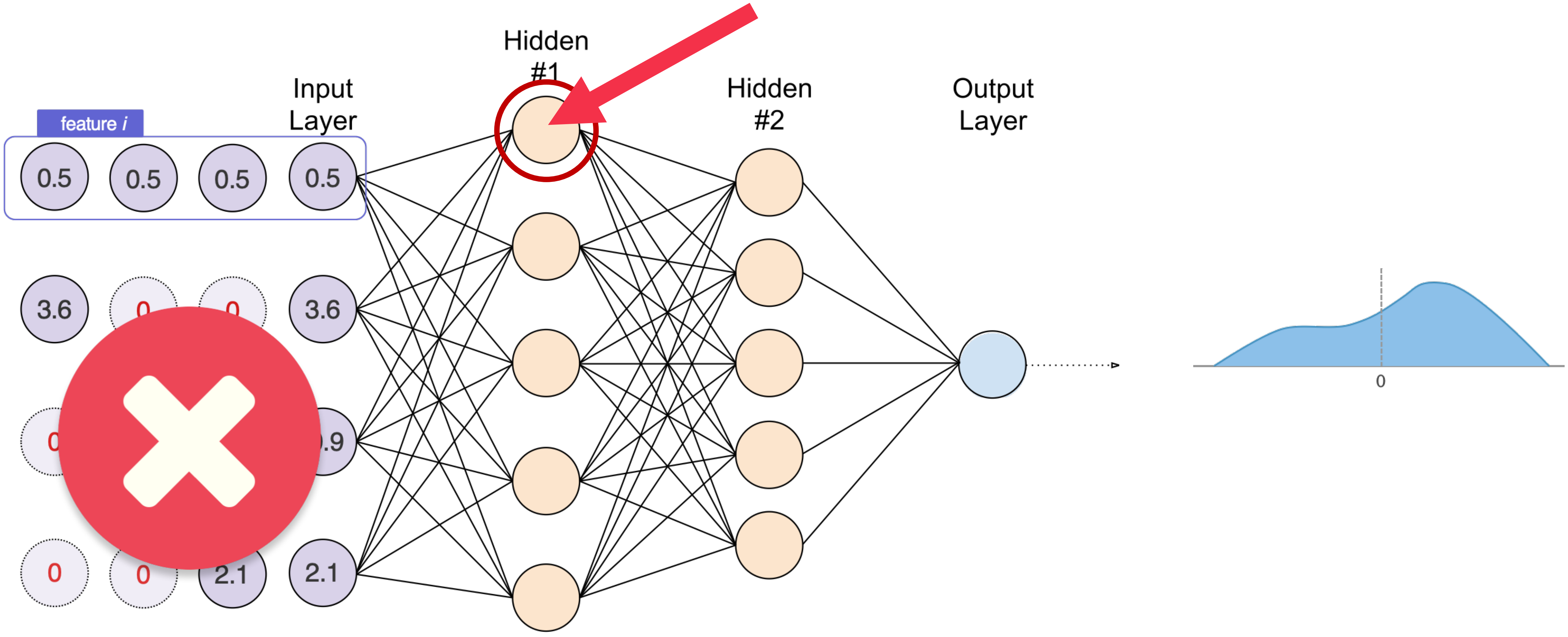
Shapley value sampling



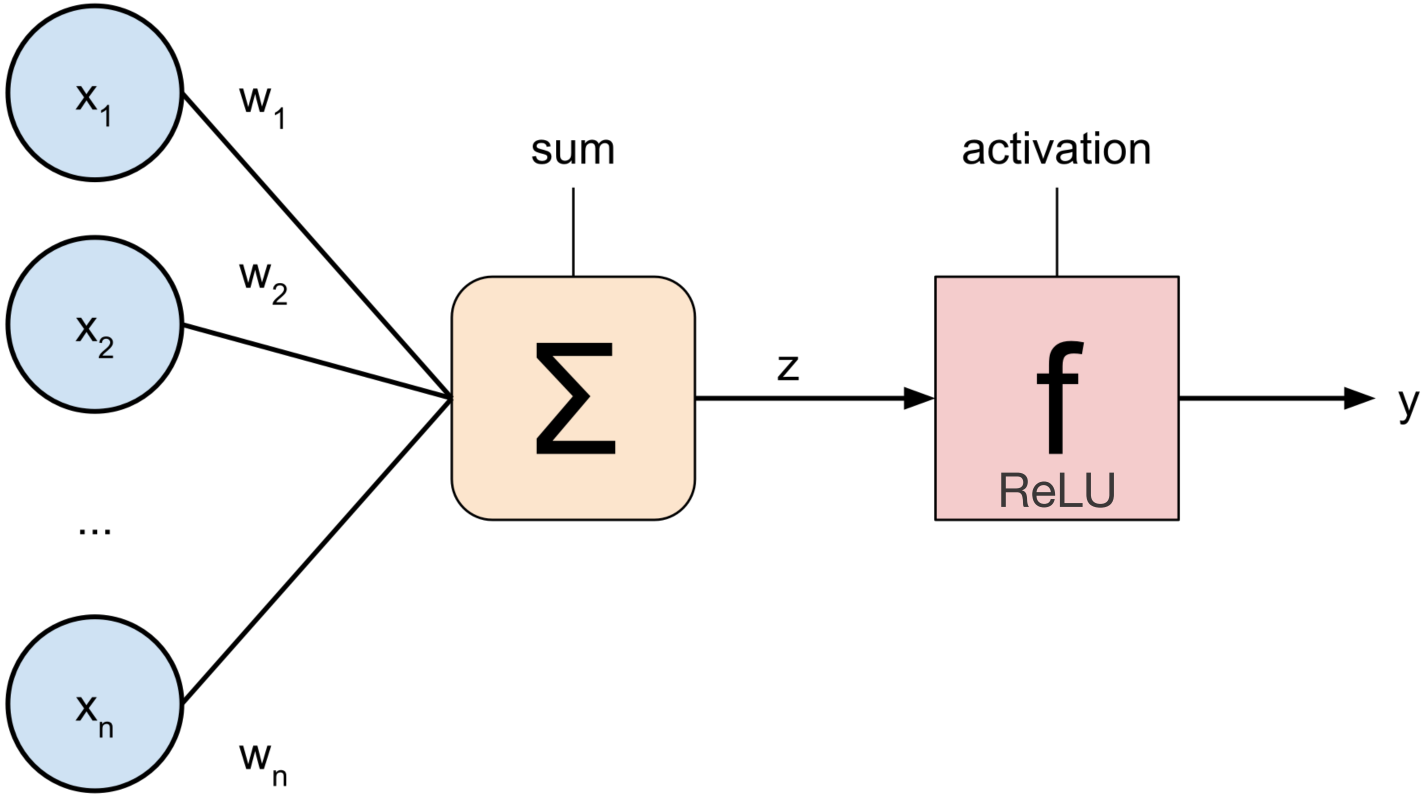
Deep Approximate Shapley Propagation



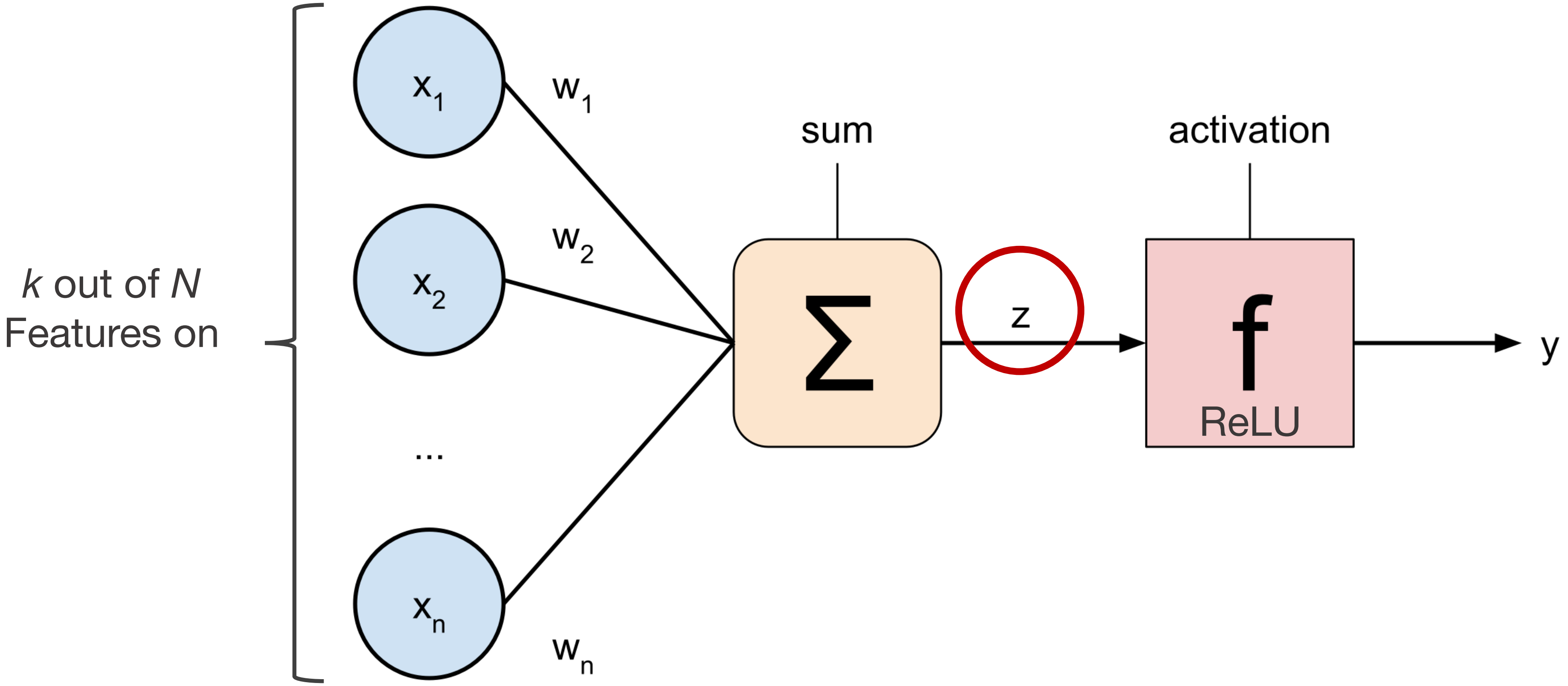
Deep Approximate Shapley Propagation



Deep Approximate Shapley Propagation



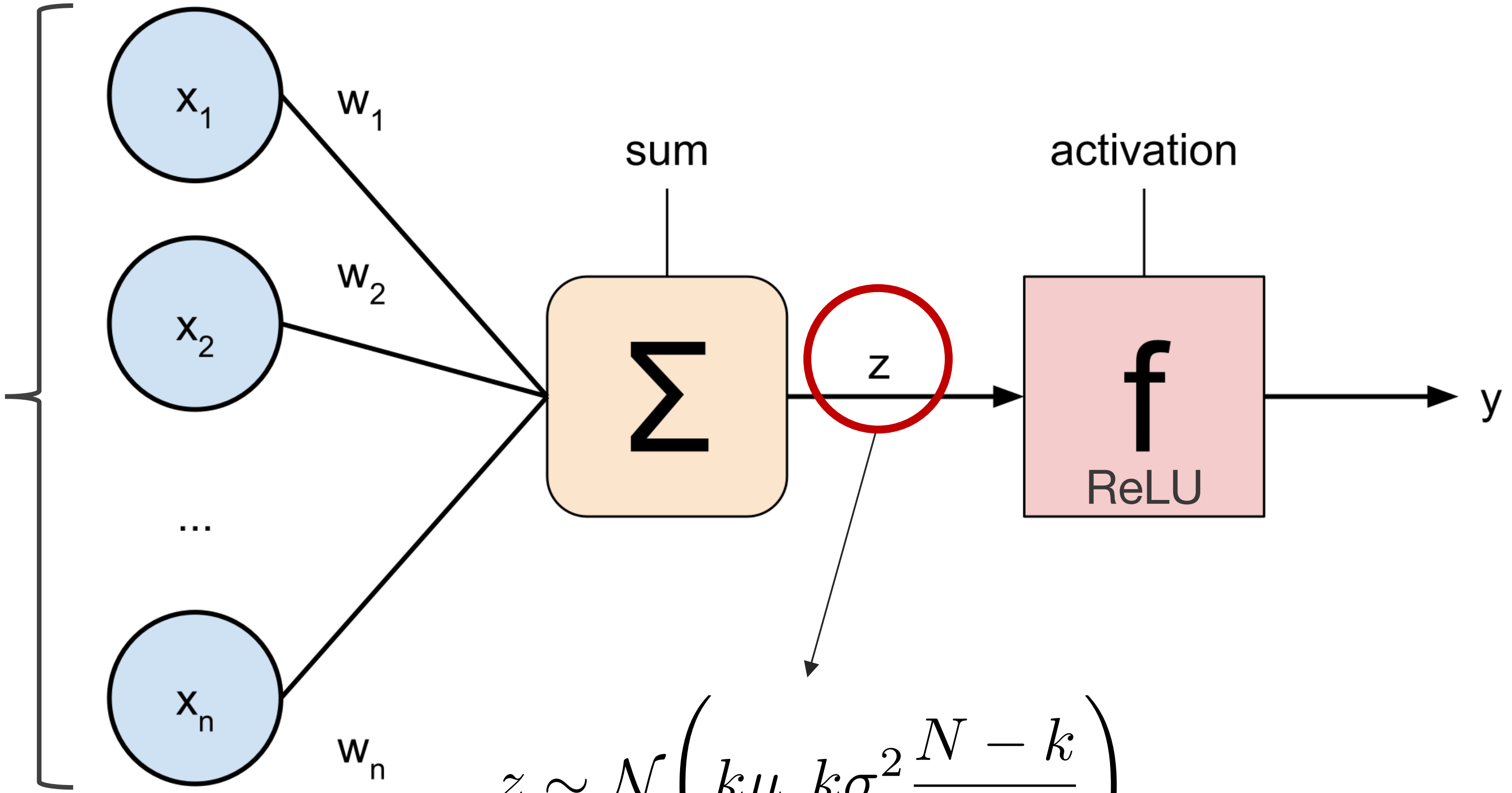
Deep Approximate Shapley Propagation



Deep Approximate Shapley Propagation

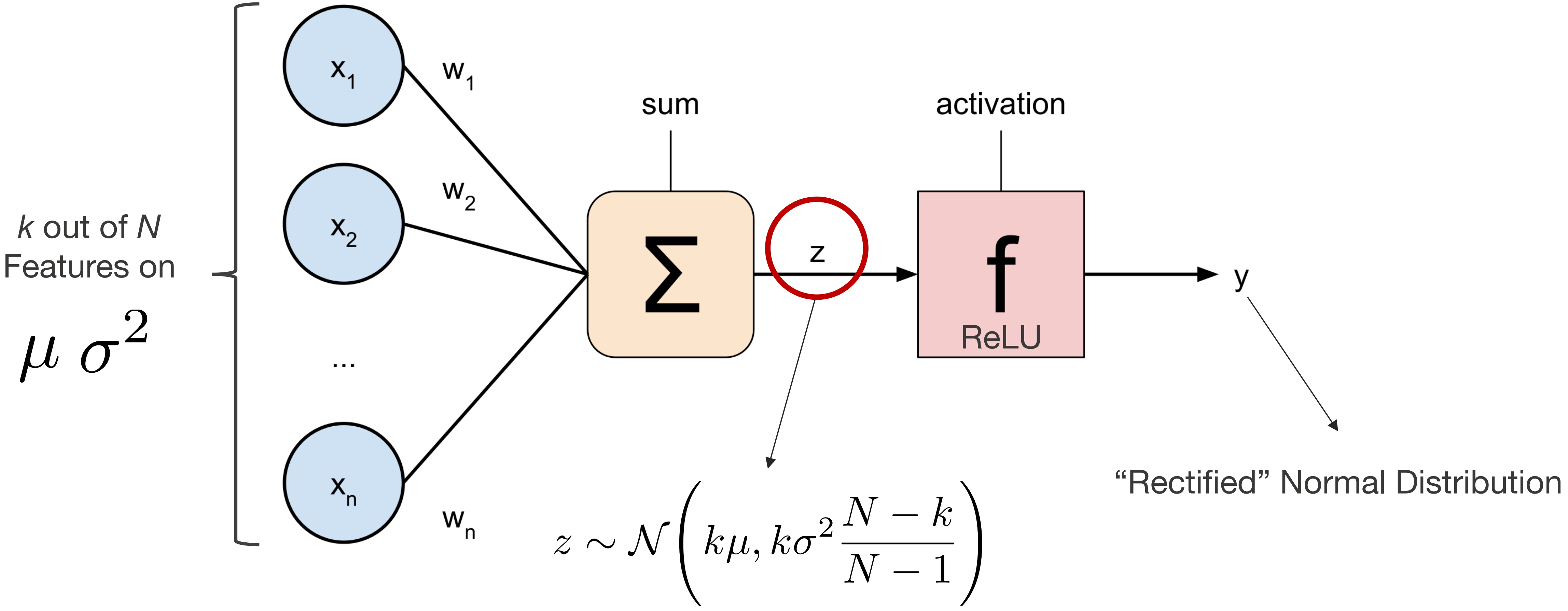
k out of *N*
Features on

$\mu \; \sigma^2$

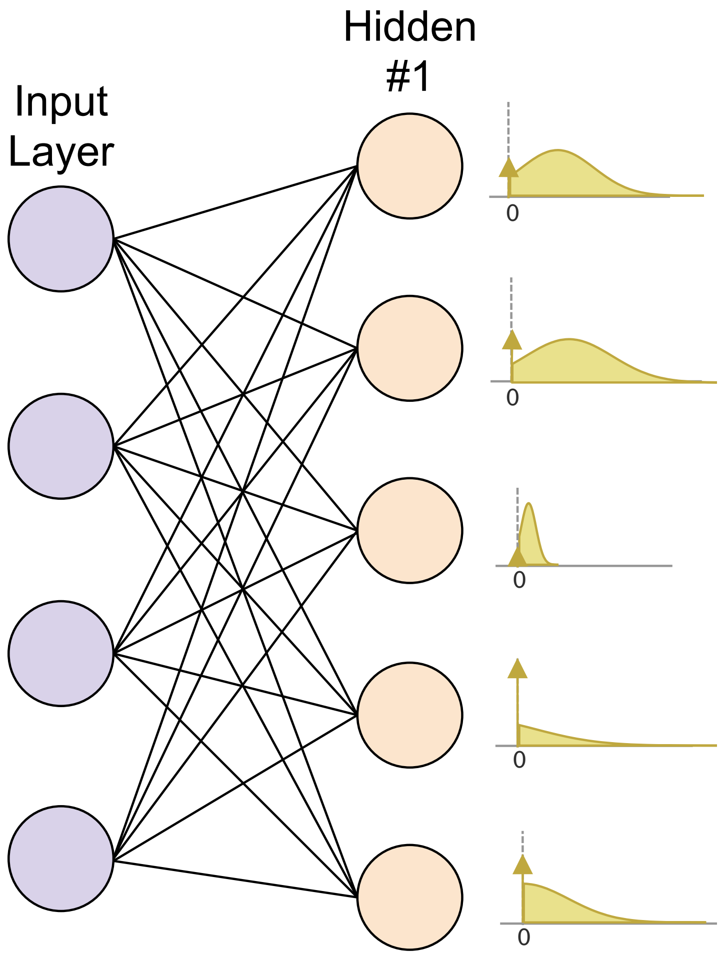


$$z \sim \mathcal{N}\left(k\mu, k\sigma^2 \frac{N-k}{N-1}\right)$$

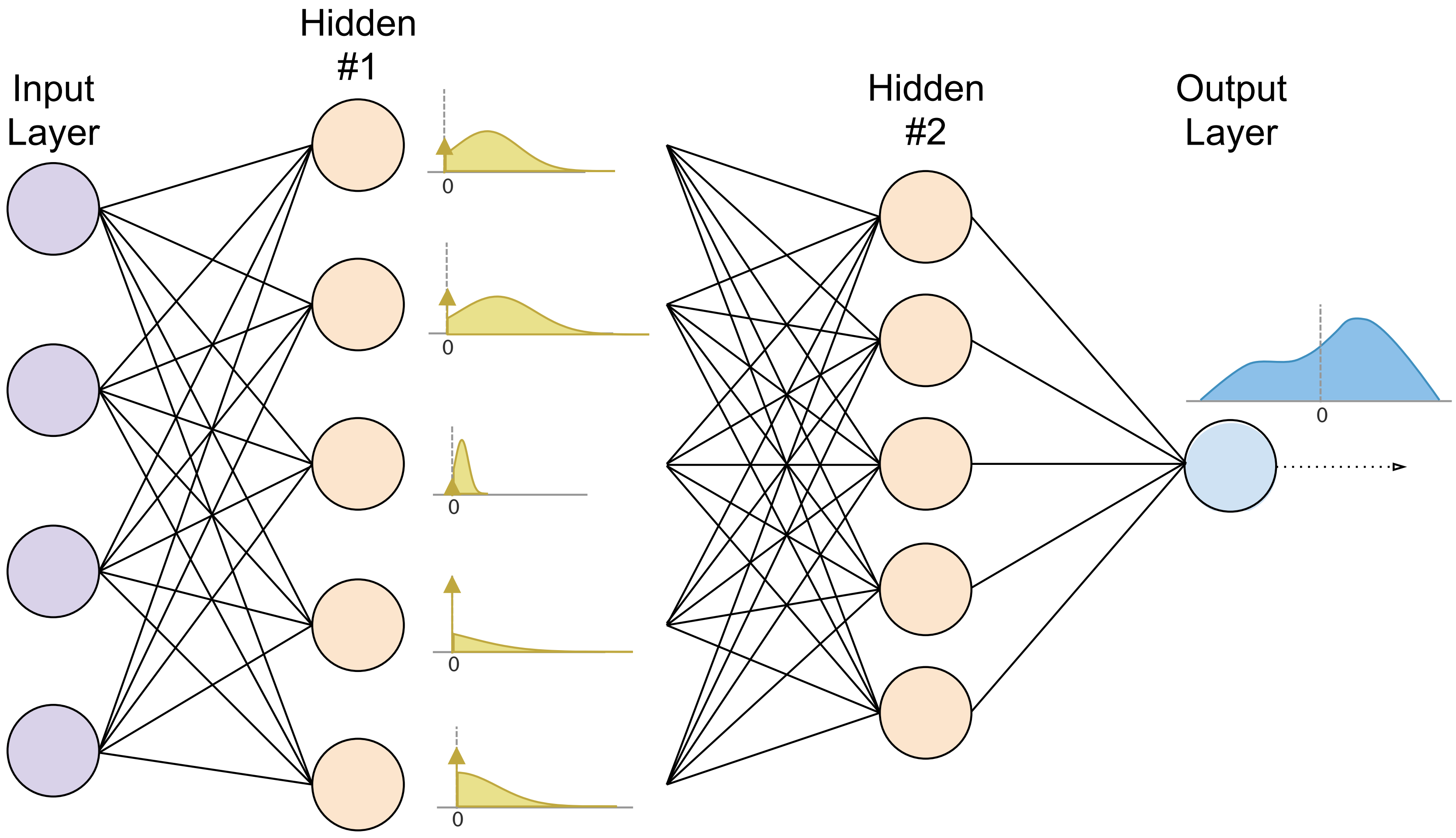
Deep Approximate Shapley Propagation



Deep Approximate Shapley Propagation



Deep Approximate Shapley Propagation



Deep Approximate Shapley Propagation

To propagate distributions through the network layers we use **Lightweight Probabilistic Deep Networks** Gast et al., 2018

Deep Approximate Shapley Propagation

To propagate distributions through the network layers we use **Lightweight Probabilistic Deep Networks** Gast et al., 2018

Affine transformation

Rectified Linear Unit

Leaky Rectified Linear Unit

Mean pooling

Max pooling

...

The use of other probabilistic frameworks is also possible

DASP vs other methods

Gradient-based methods

- ✓ (Very) fast
- × Poor Shapley Value estimation

DASP

Sampling-based methods

- ✓ Unbiased Shapley Value estimator
- × Slow

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Disney RESEARCH
STUDIOS

For details, come at the poster
Pacific Ballroom #63

Lightweight Probabilistic Deep Network (Keras)
github.com/marcoancona/LPDN

Deep Approximate Shapley Propagation
github.com/marcoancona/DASP

Thank you

References

Lloyd S. Shapley, A value for n-person games, 1952
Castro et al., Polynomial calculation of the Shapley value based on sampling, 2009
Fatima et al., A linear approximation method for the Shapley value, 2014
Ribeiro et al., "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016
Sundararajan et al., Axiomatic attribution for deep networks, 2017
Shrikumar at al., Learning important features through propagating activation differences, 2017
Lundberg et al., A Unified Approach to Interpreting Model Predictions, 2017
Gast et al., Lightweight Probabilistic Deep Networks, 2018