A glance inside the black box

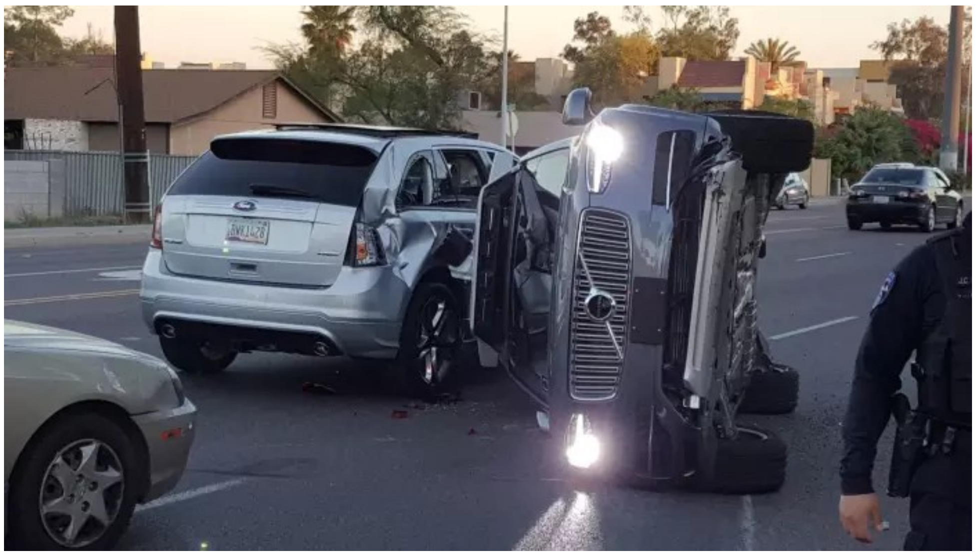
Deep Learning @ MLHEP 2019

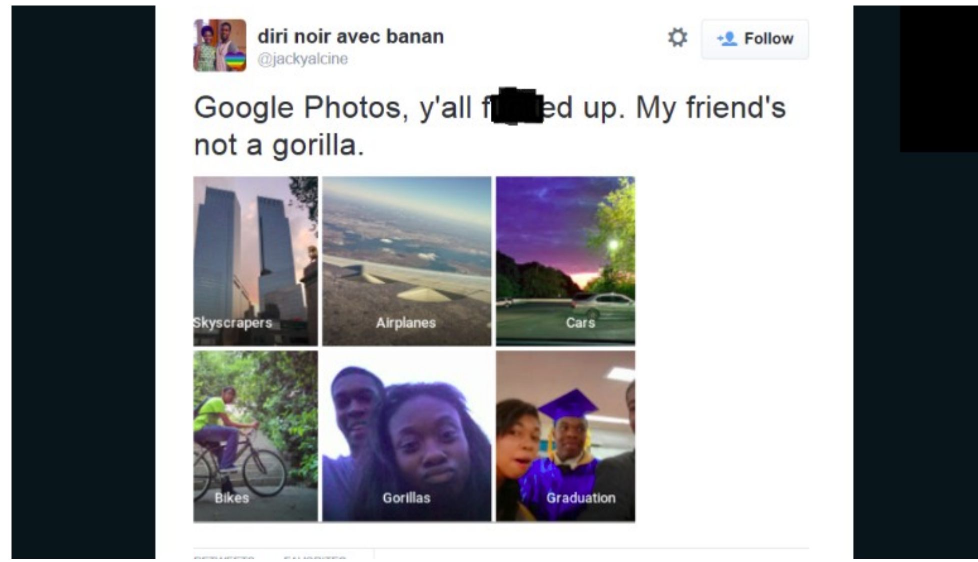
Yandex
Research

LAMBDA

British Hedgehog
Preservation Society

ML mistakes have a cost

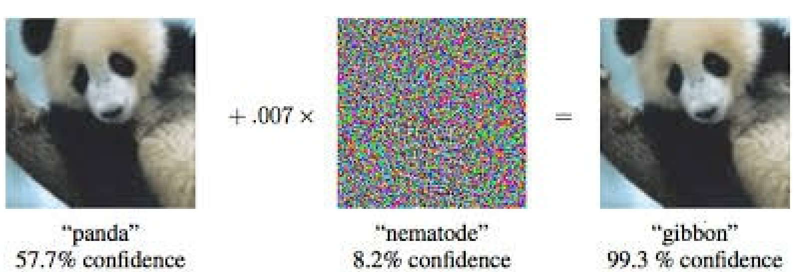




Uber self-driving car crashes dur
US tests

3. Robot injured a child

A so-called "crime fighting robot," created by the platform
into a child in a Silicon Valley mall in July, injuring the 16-



Chinese billionaire's face identified as jaywalker

Traffic police in major Chinese cities are using AI to address jaywalking.
They deploy smart cameras using facial recognition techniques at

The question of trust

How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

The question of trust

How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

Is my model certain about what it says?
Is there something wrong with this input?
Can I rely on this prediction?

The question of trust

How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

Is my model certain about what it says?
Is there something wrong with this input?
Can I rely on this prediction?

Can I trust this data?
Is something missing?
Is there any bias?

The question of trust

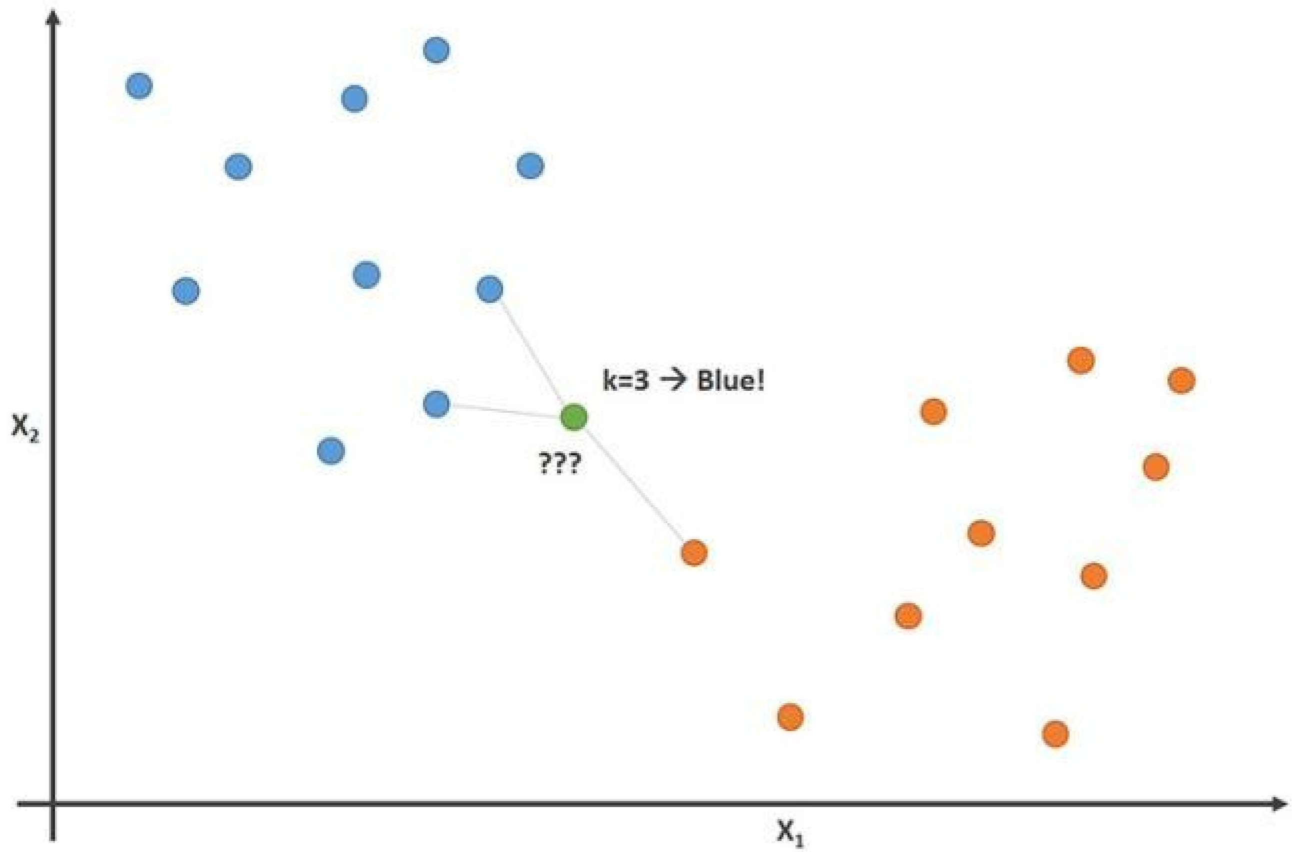
How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

Is my model certain about what it says?
Is there something wrong with this input?
Can I rely on this prediction?

Can I trust this data?
Is something missing?
Is there any bias?

What is interpretable?

Simple stuff like **K Nearest Neighbors**



What is interpretable?

Simple stuff like **Linear models**

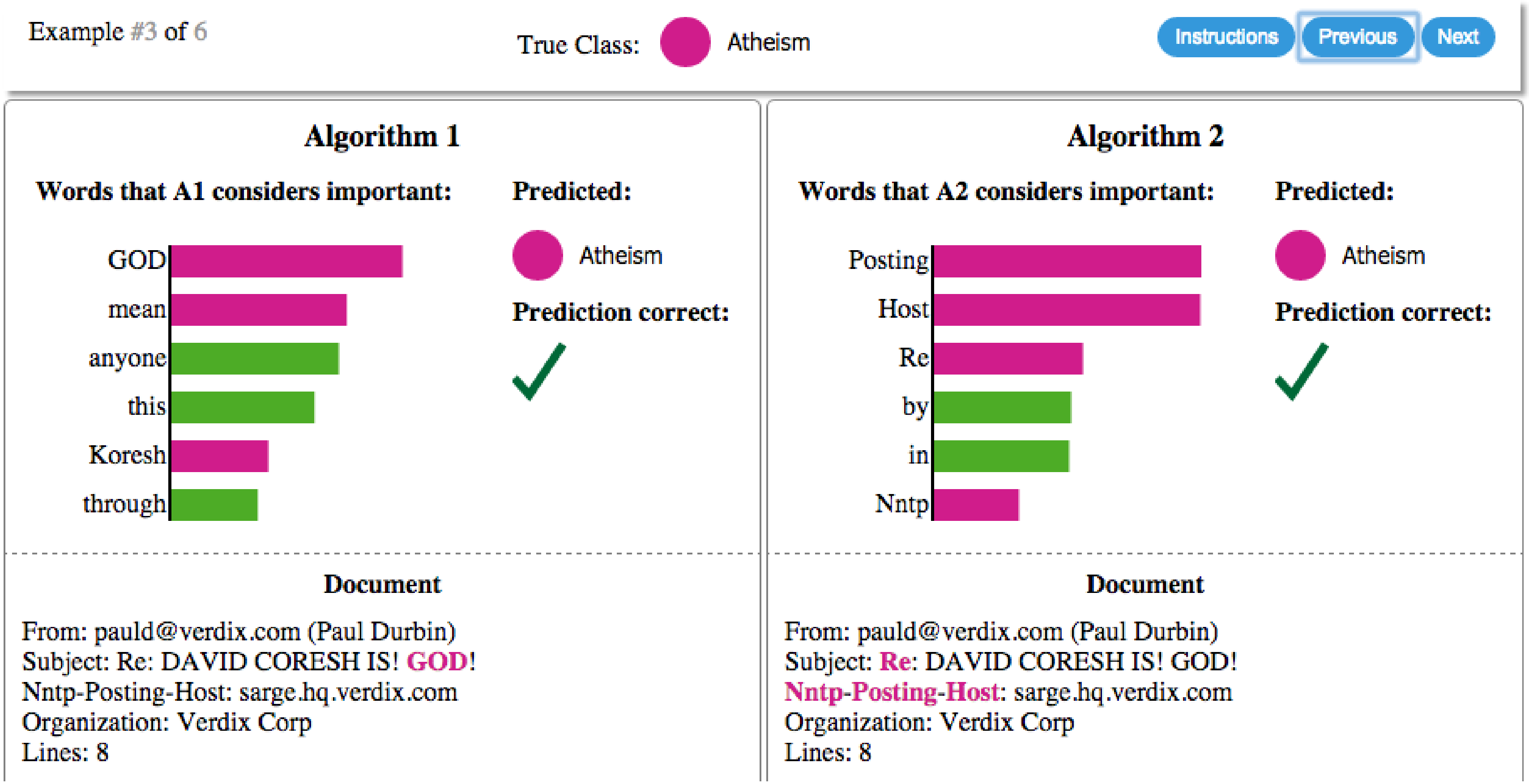
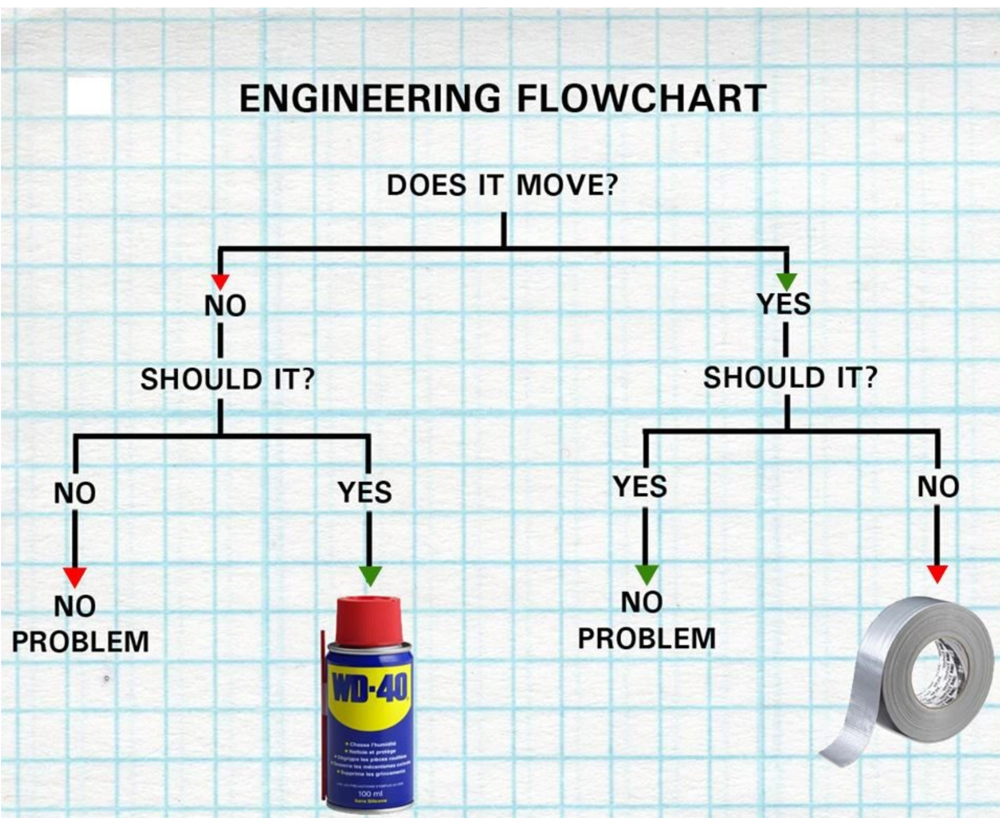
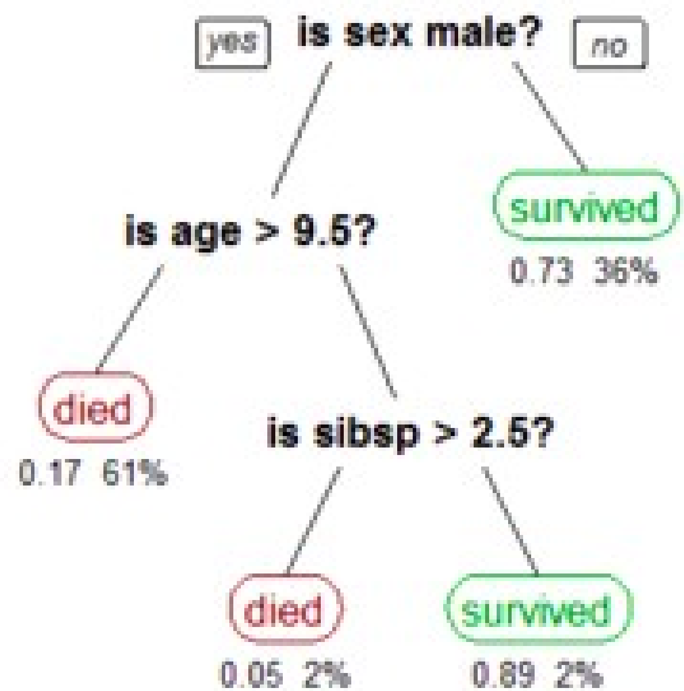


Image: weights of words in linear text classifier

What is interpretable?

Simple stuff like **Decision Trees**

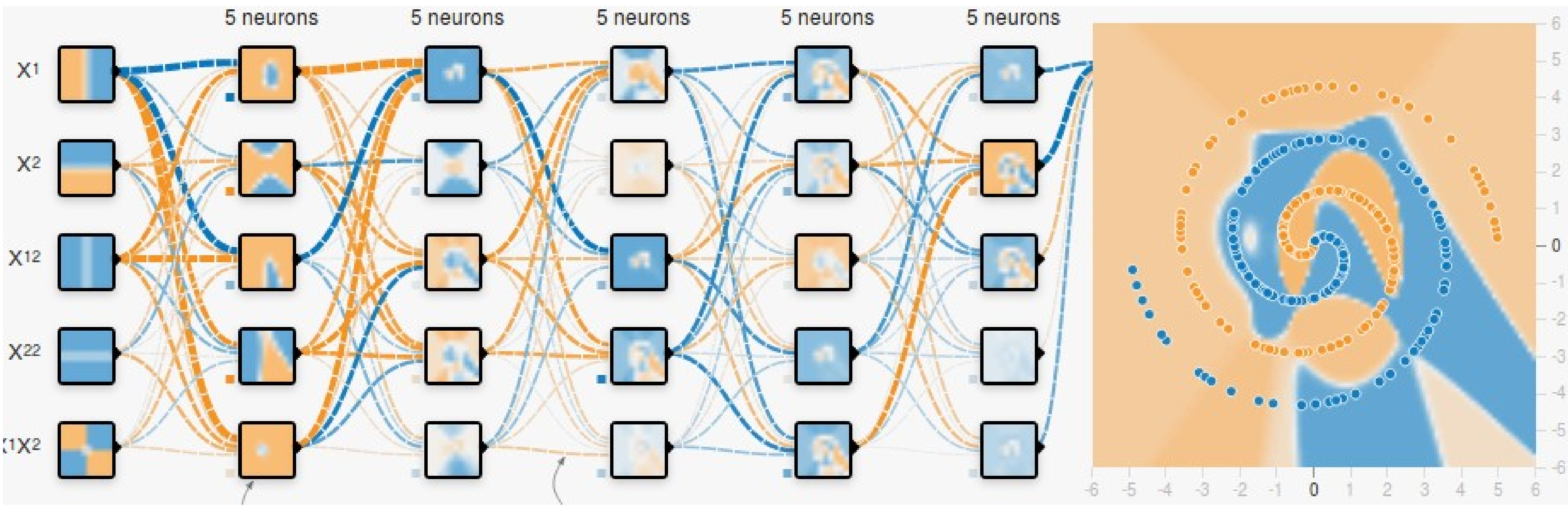




Survival on Titanic

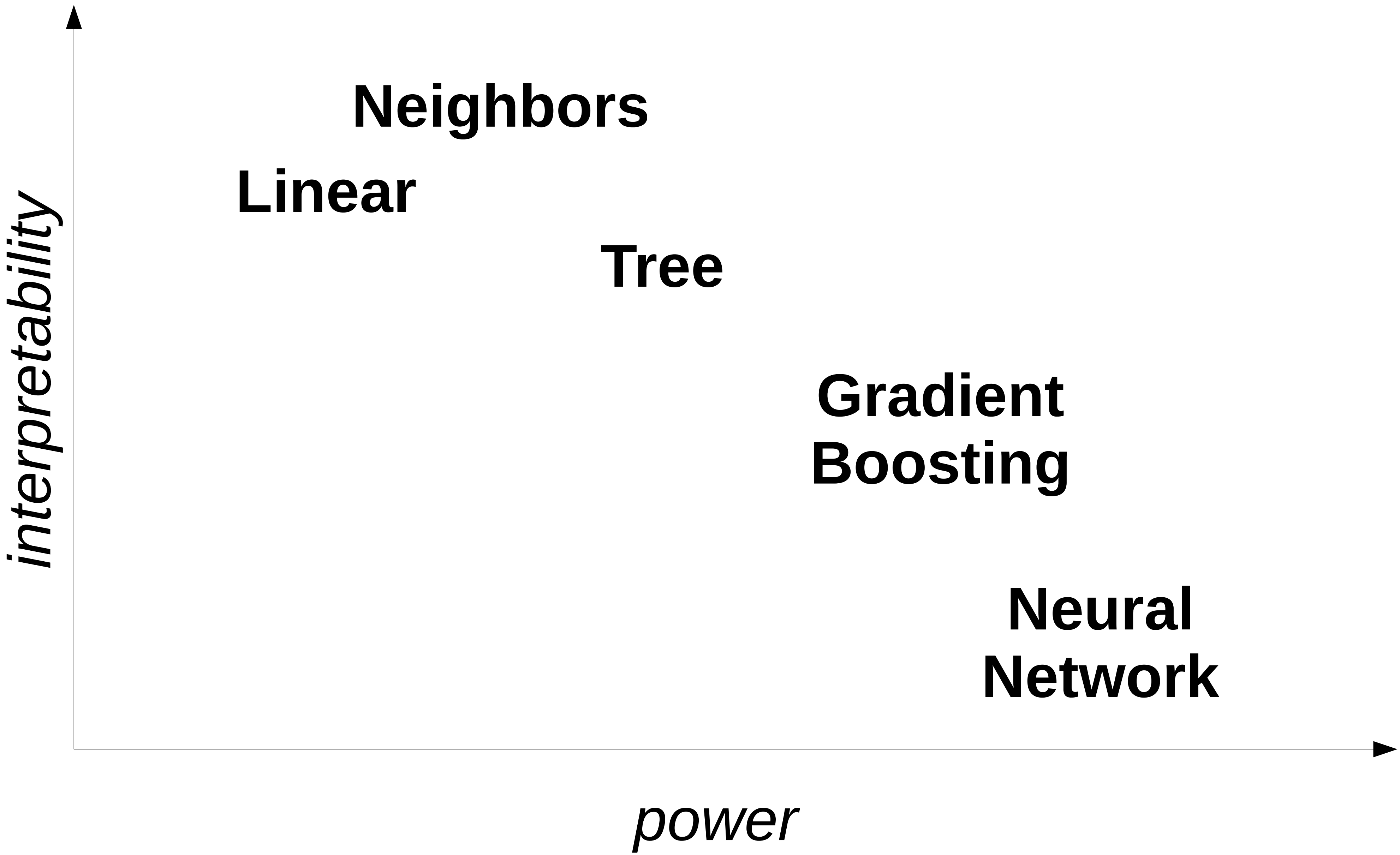
What is interpretable?

Neural networks are **not** naturally interpretable

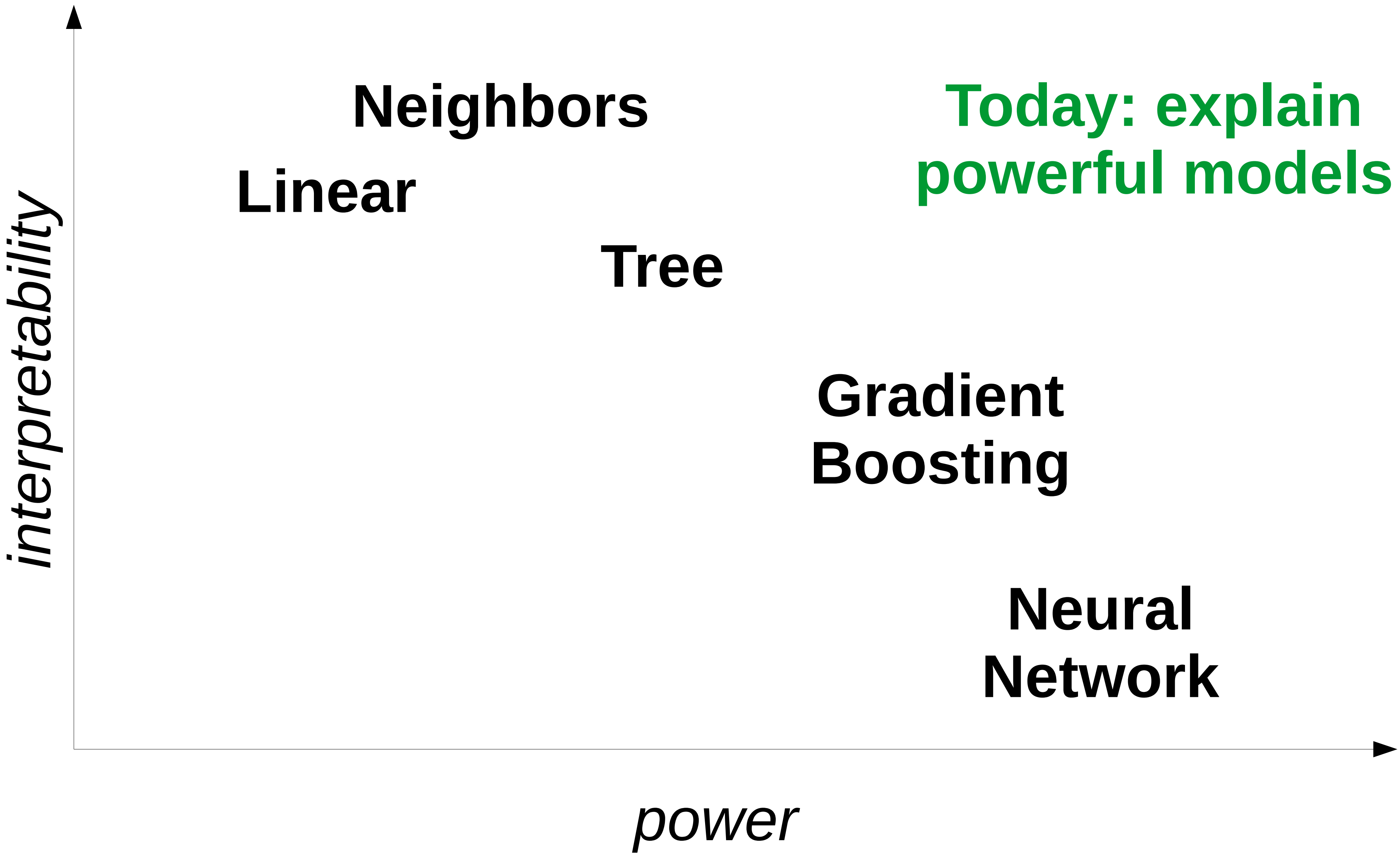


Source: https://playground.tensorflow.org

Power vs interpretability



Power vs interpretability

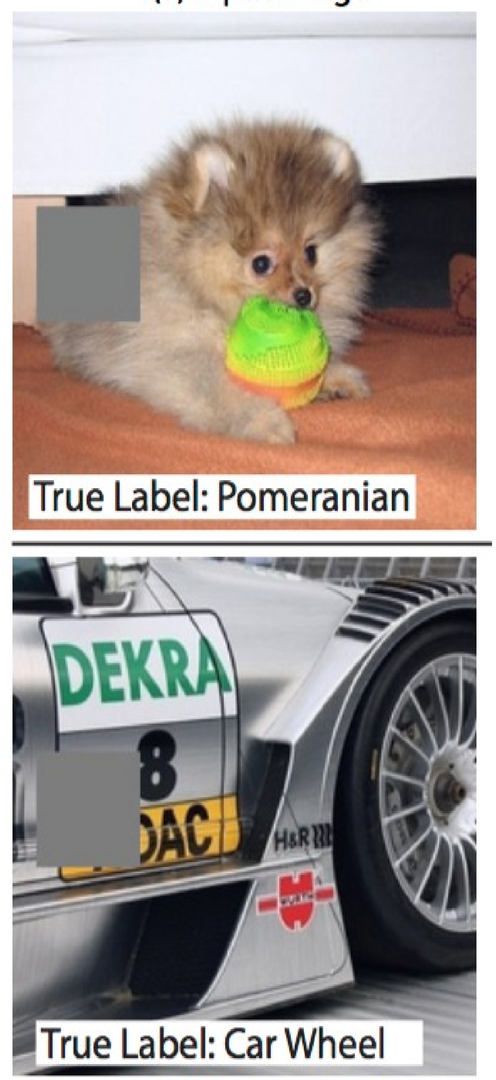


Explanation by occlusion

Idea:

- Let’s add noise to inputs and see what happens!
- For images: slide a gray square over the image, measure how it affects predictions

(a) Input Image



(d) Classifier, probability of correct class

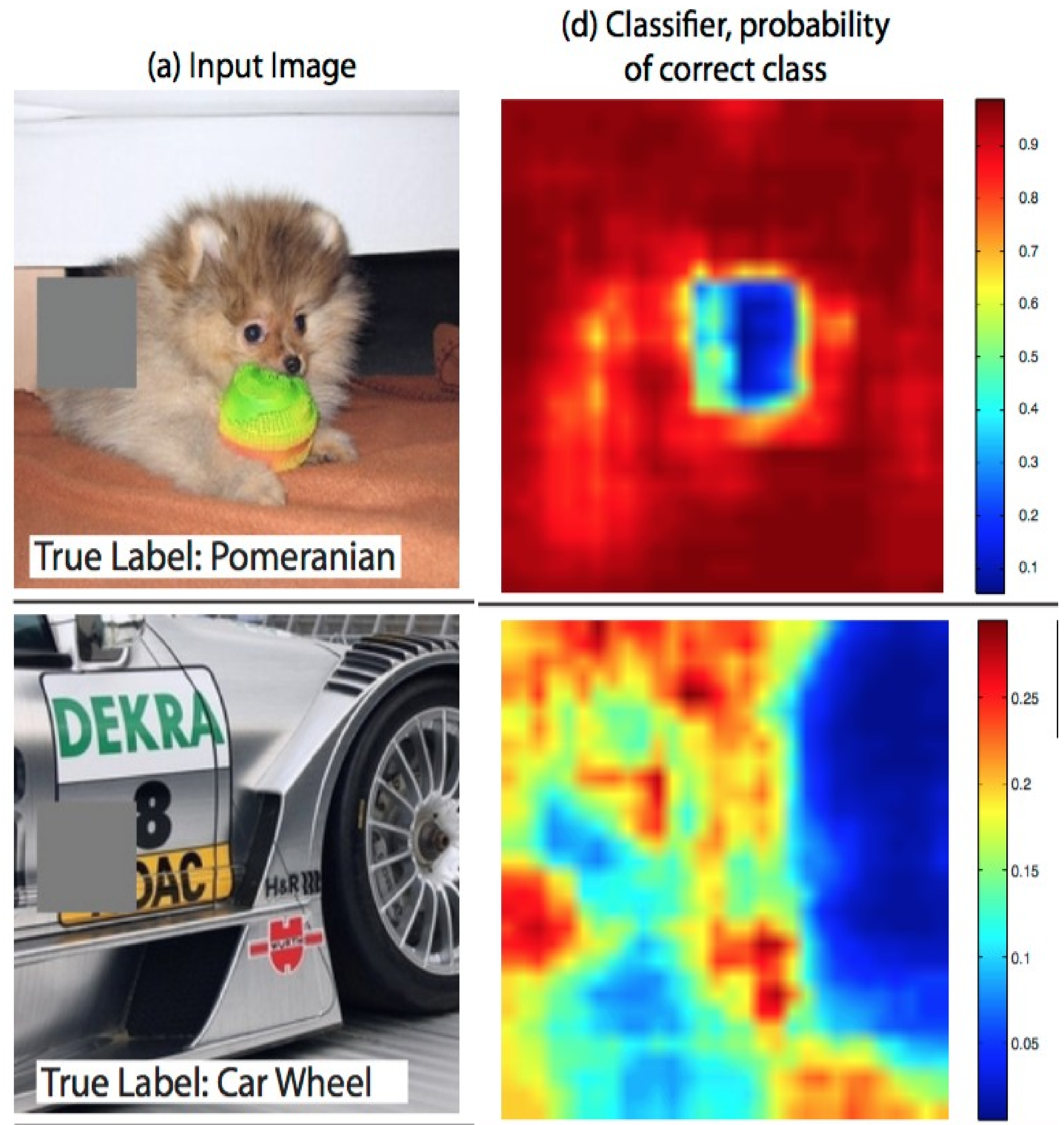
Your guess?

Explanation by occlusion

Idea:

- Let's add noise to inputs and see what happens!

- For images: slide a gray square over the image, measure how it affects predictions



Explanation by occlusion

Idea:

- Let’s add noise to inputs and see what happens!

- For texts: drop individual words and measure how it affects predictions

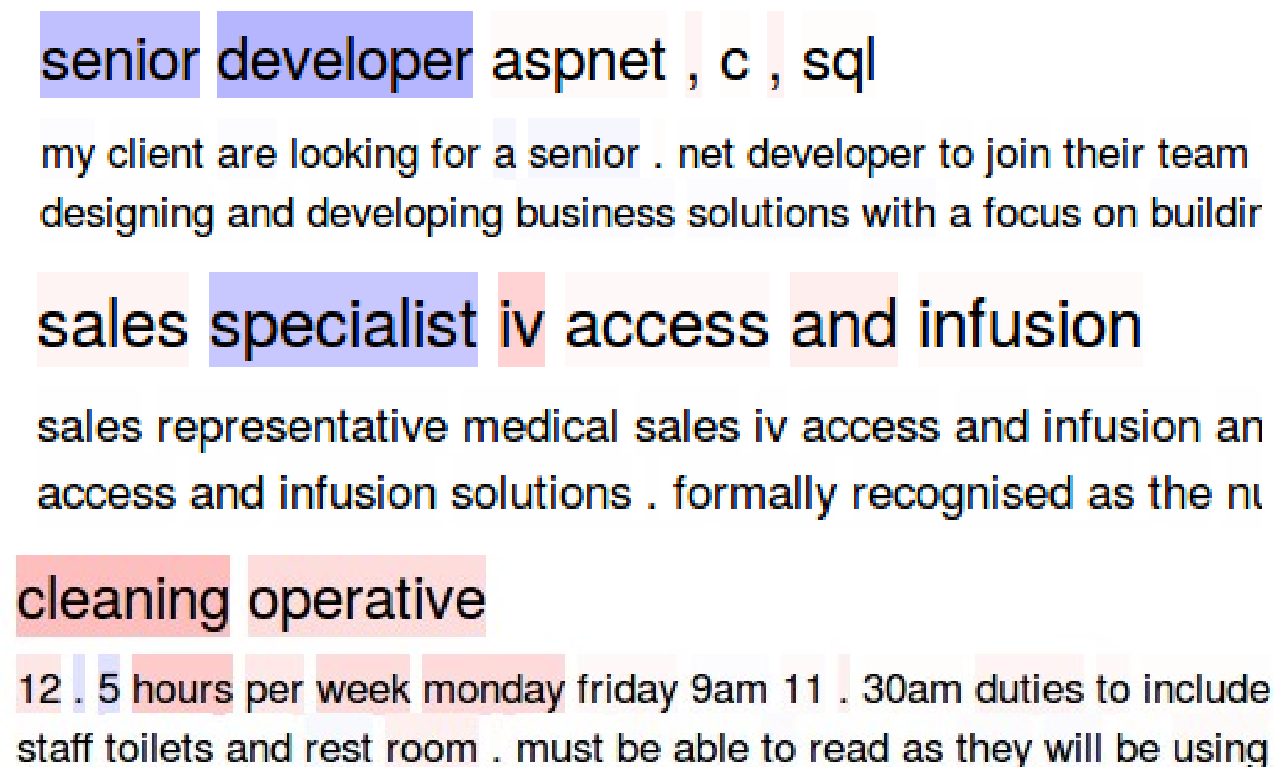


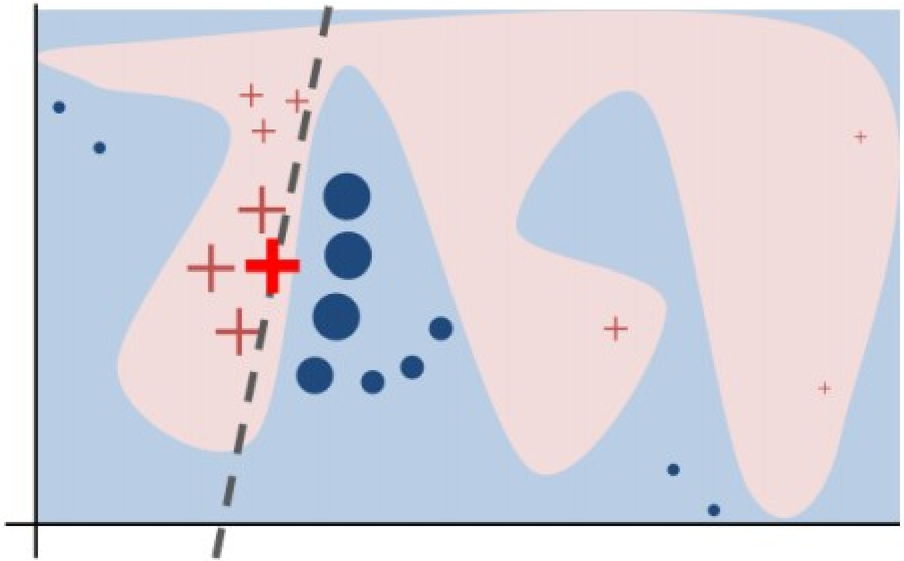
Image: salary prediction

Explanation by approximation

Idea:

- Approximate your model with something explainable
  e.g. linear model

- The approximation only needs to hold **locally**
  i.e. on similar inputs



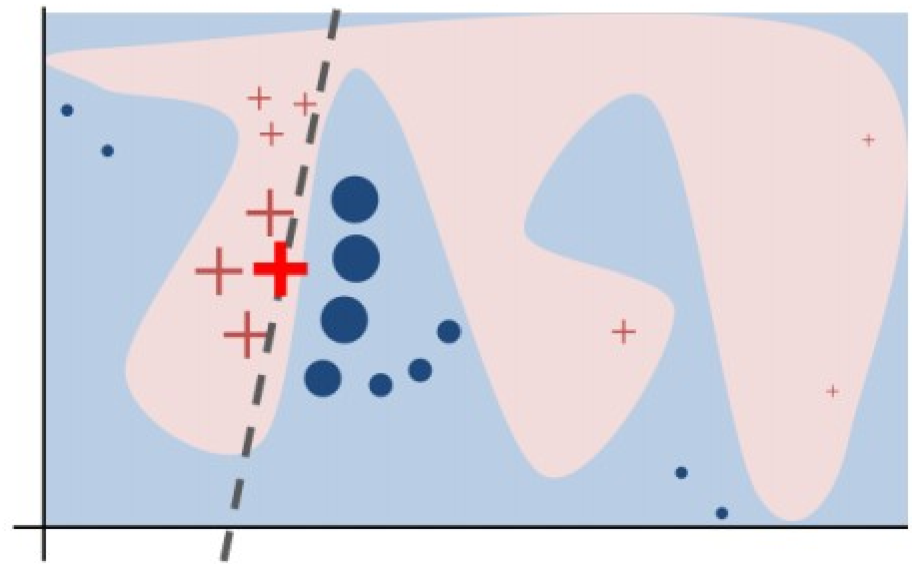
Read more in the paper

Explanation by approximation

Idea:

- Approximate your model with something explainable
  e.g. linear model

- The approximation only needs to hold **locally**
  i.e. on similar inputs



Read more in the paper

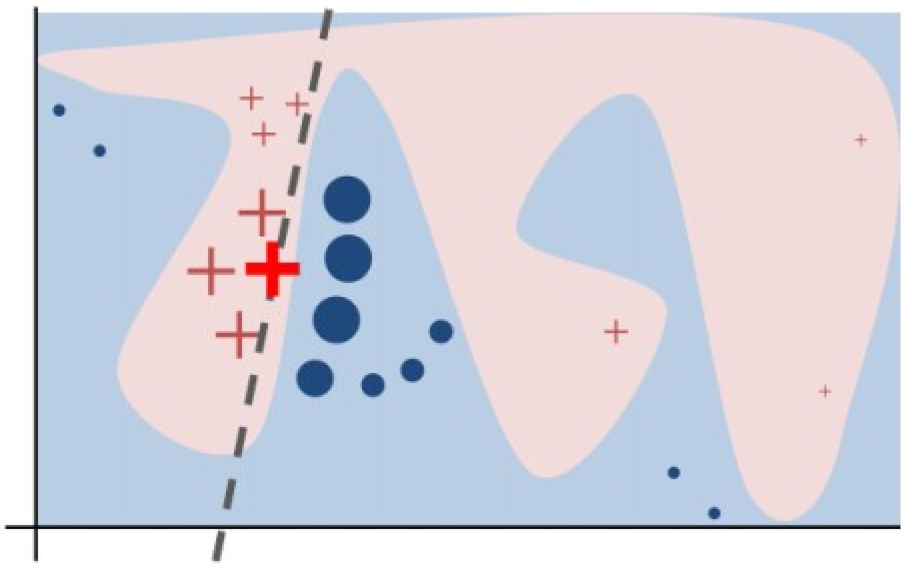
(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

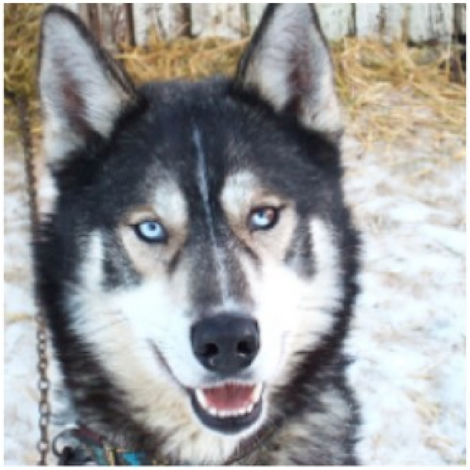
Explanation by approximation

Idea:

- Approximate your model with something explainable
  e.g. linear model

- The approximation only needs to hold **locally**
  i.e. on similar inputs





(a) Husky classified as wolf

(b) Explanation

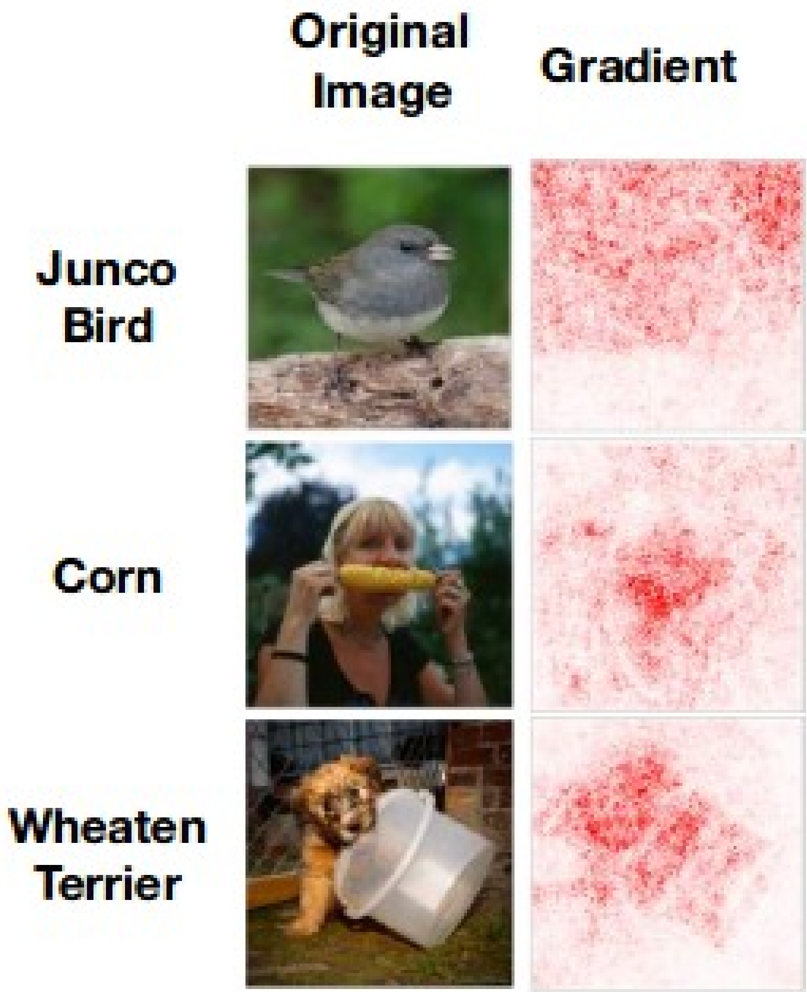
Left image: model mislabeled a husky dog as a wolf; explanation: snow :)

Figures taken from the paper
Why Should I Trust You?

Explanation by gradients

Idea: use gradients!

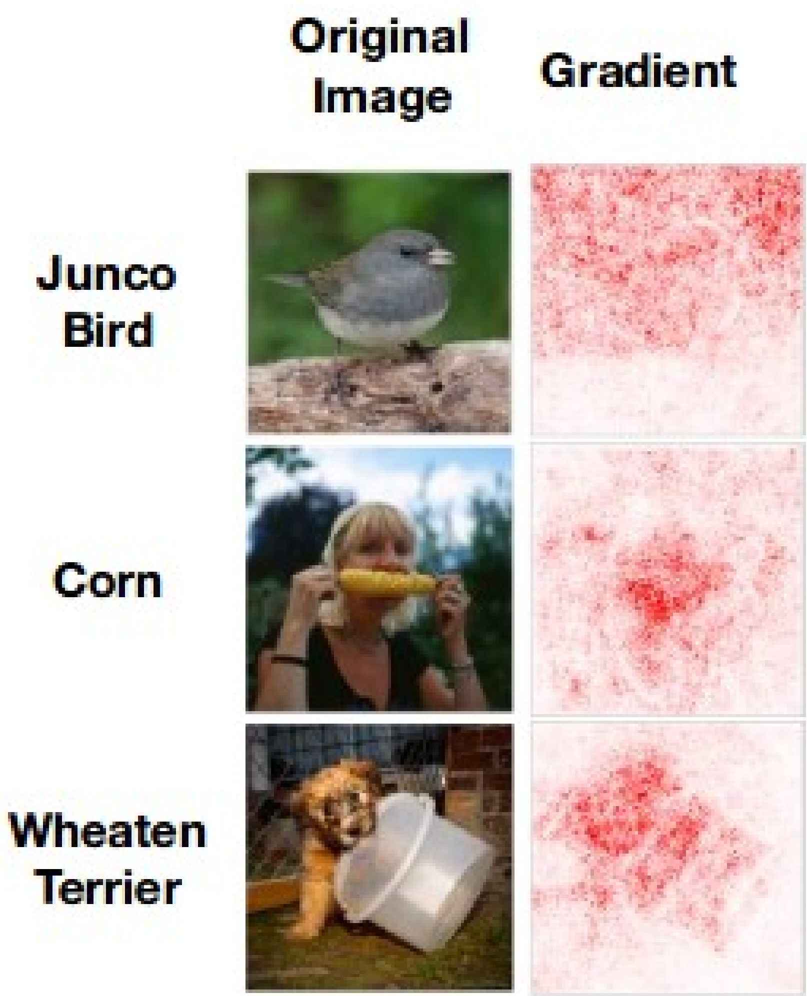
$$\nabla_{x_i} model(x) = \frac{\partial\, model(x)}{\partial x_i}$$



Explanation by gradients

Idea: use gradients!

$$\nabla_{x_i} model(x) = \frac{\partial\, model(x)}{\partial\, x_i}$$



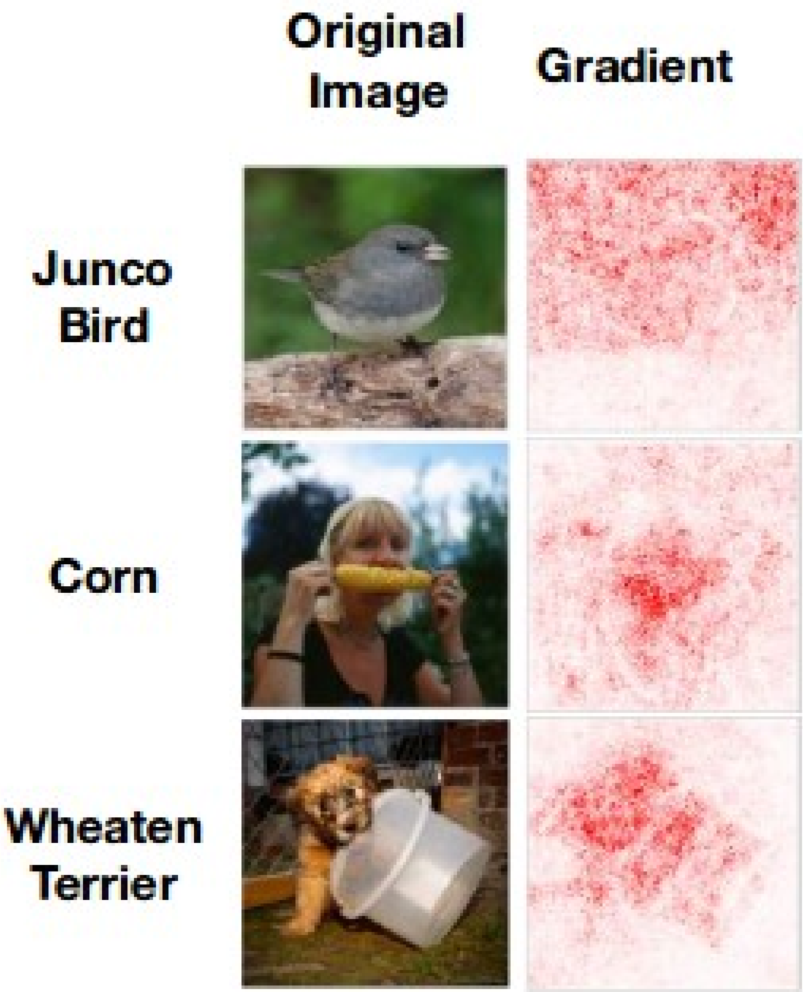
Gradients are too sensitive to small changes in **x**

Q: How would you fix that?

Explanation by gradients

Idea: use gradients!

$$\nabla_{x_i} model(x) = \frac{\partial\, model(x)}{\partial x_i}$$



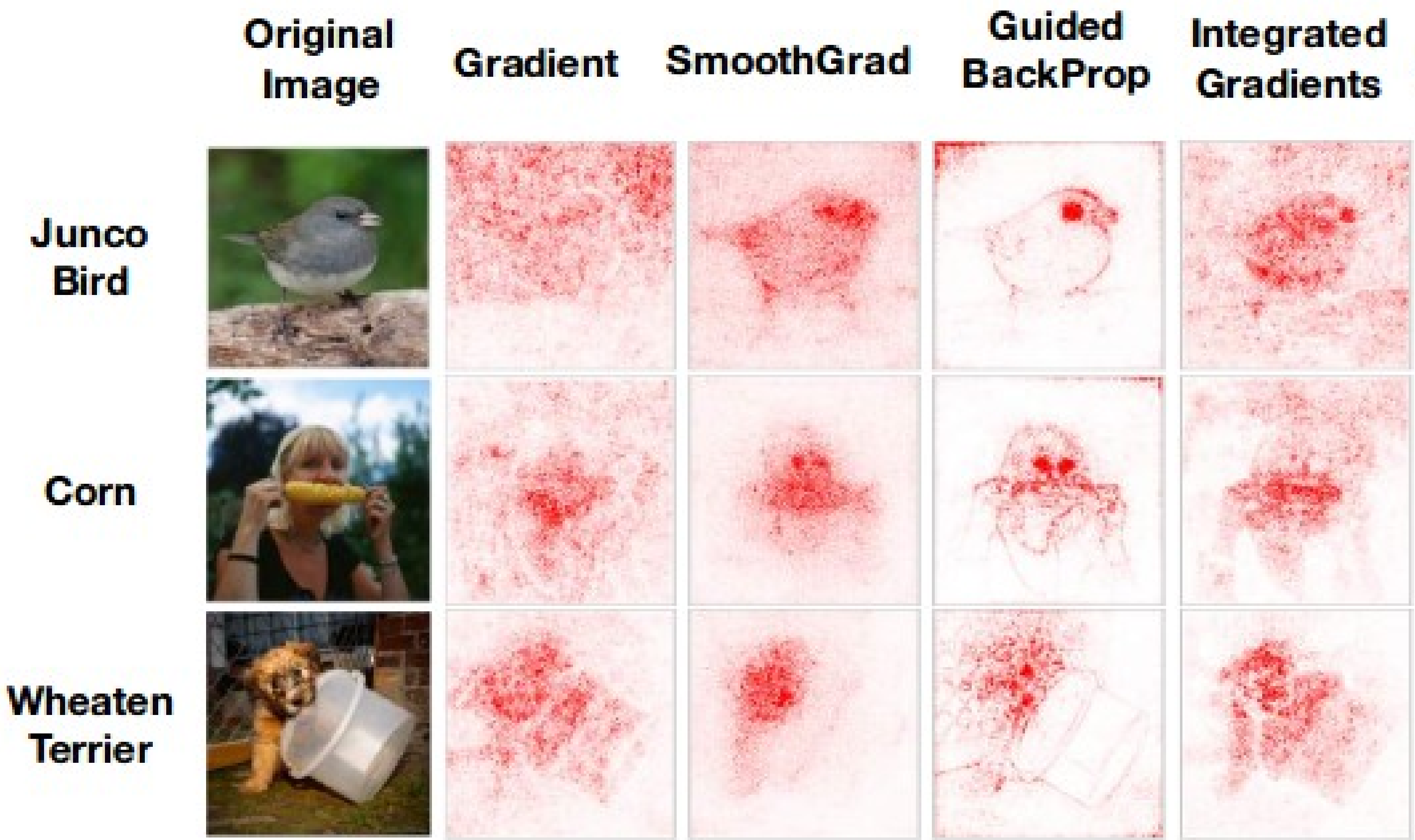
Gradients are too sensitive to small changes in **x**

Smoothgrad: average gradients over several **noisy** copies of **x**

(one of many heuristics)

Explanation by gradients

Idea: use gradients! $$\nabla_{x_i} model(x) = \frac{\partial\, model(x)}{\partial\, x_i}$$



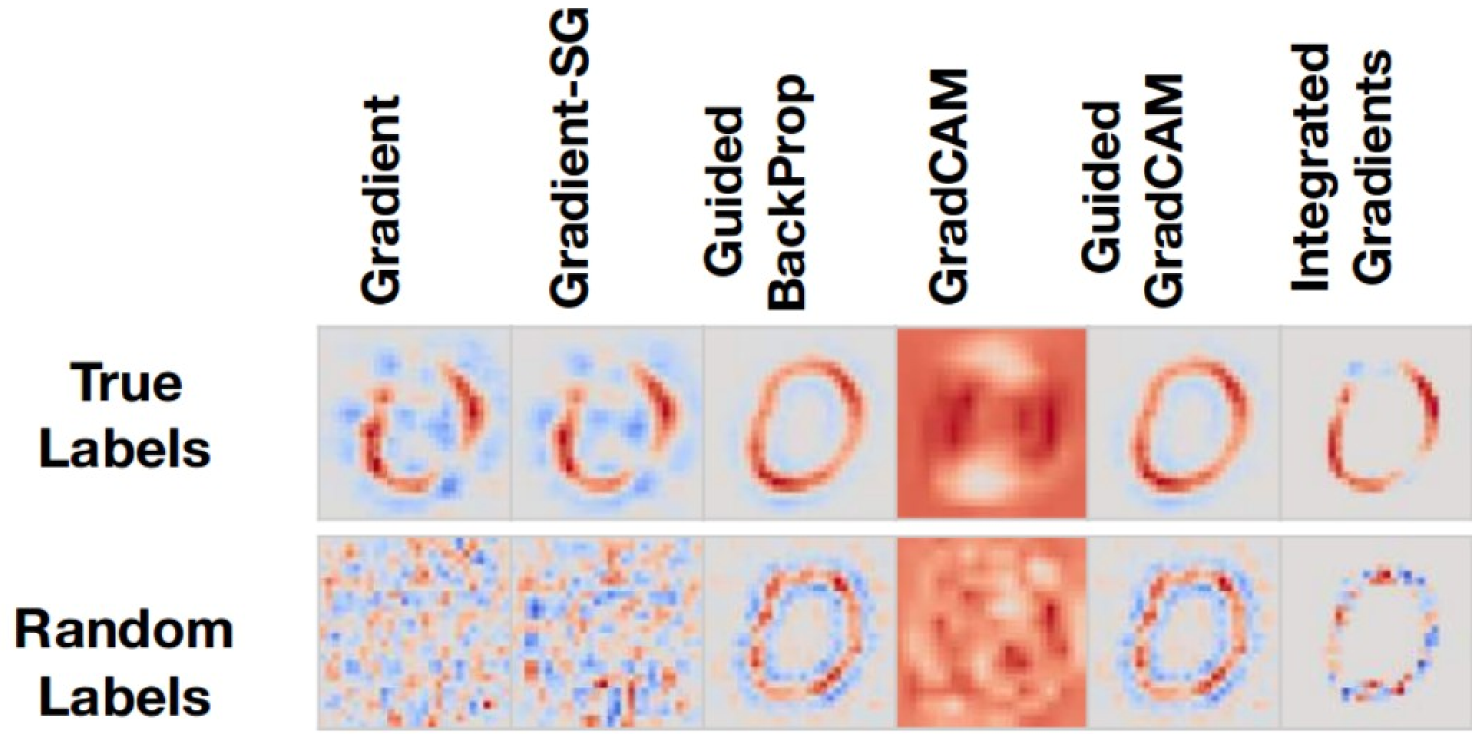
Don’t trust yourself!

The method outputs a noisy image
you see something reasonable
should you be satisfied?

How can you **verify** the explanation?

Don’t trust yourself!

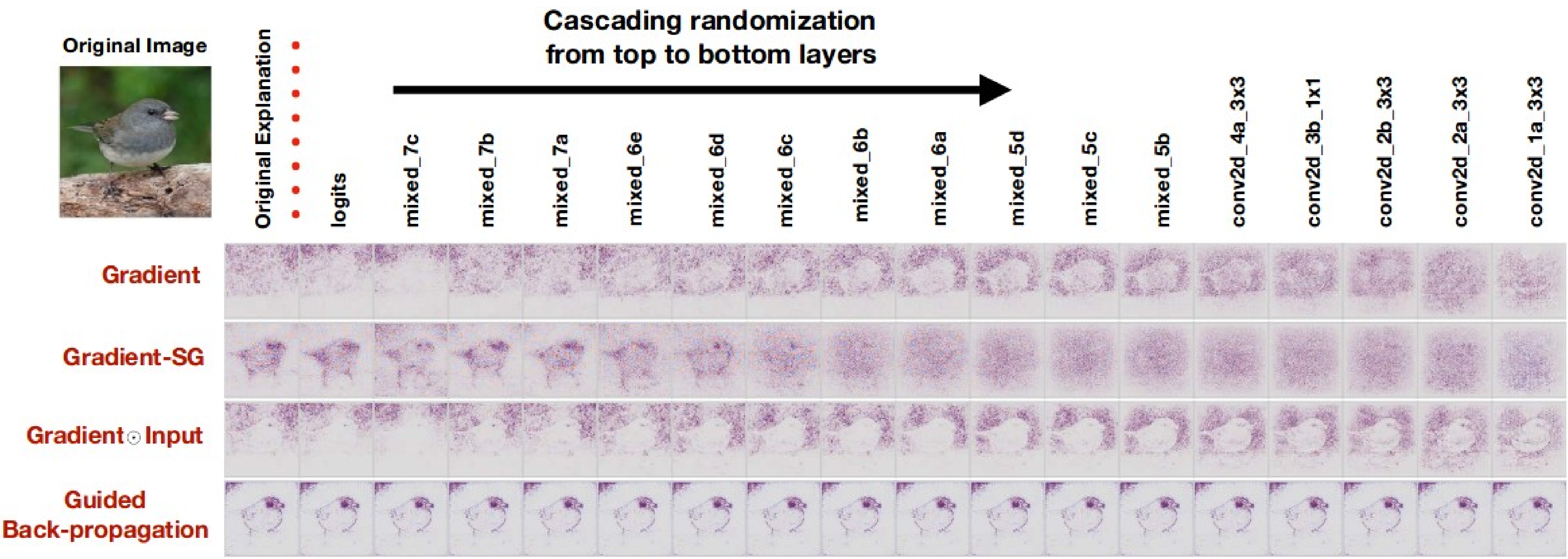
Idea: train a bogus model to see
if the method can “explain” the fake model



Source: Sanity Checks for Saliency Maps

Don’t trust yourself!

Idea: replace weights with random one layer at a time (top to bottom)

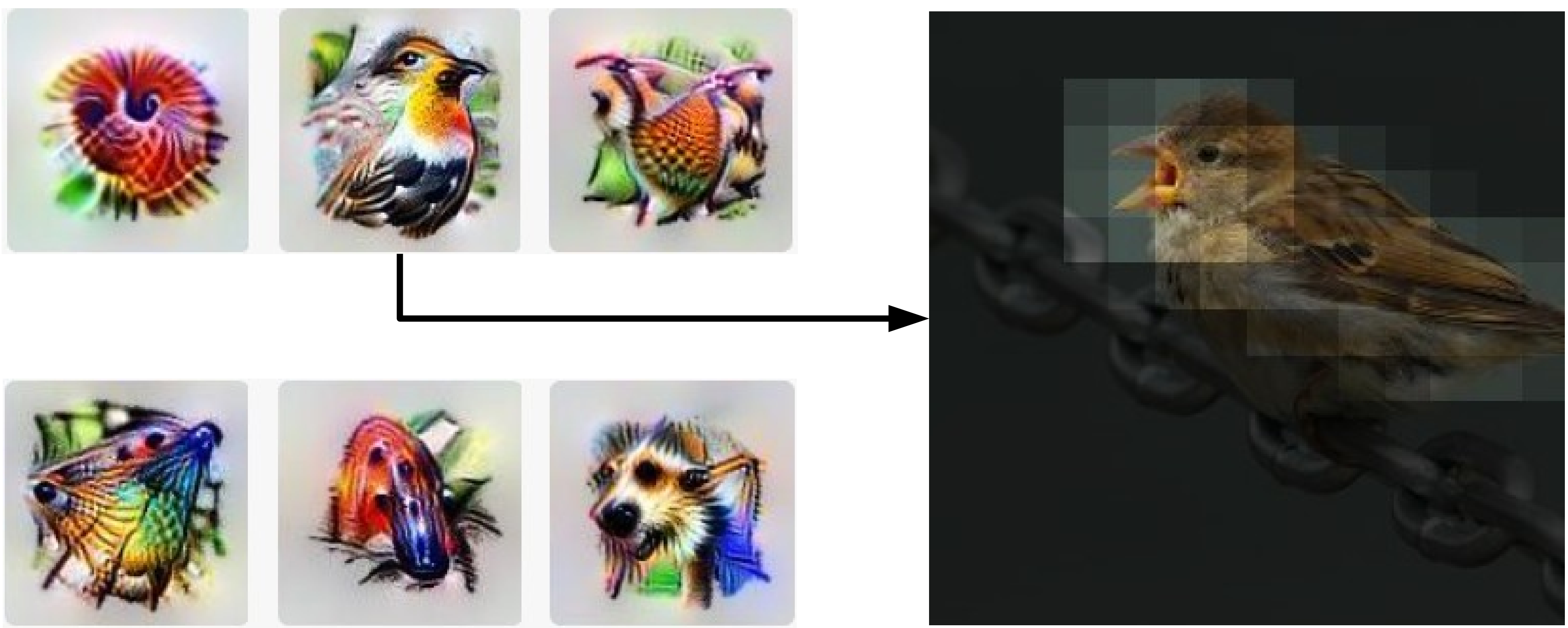


Source: *Sanity Checks for Saliency Maps*

Explanation by optimization

Idea: build an image that maximizes the activation of a particular neuron

Must read: distill.pub/2018/building-blocks



Explanation by optimization

Idea: build an image that maximizes
the activation of a particular neuron

Must read: **distill.pub/2018/building-blocks**

More:

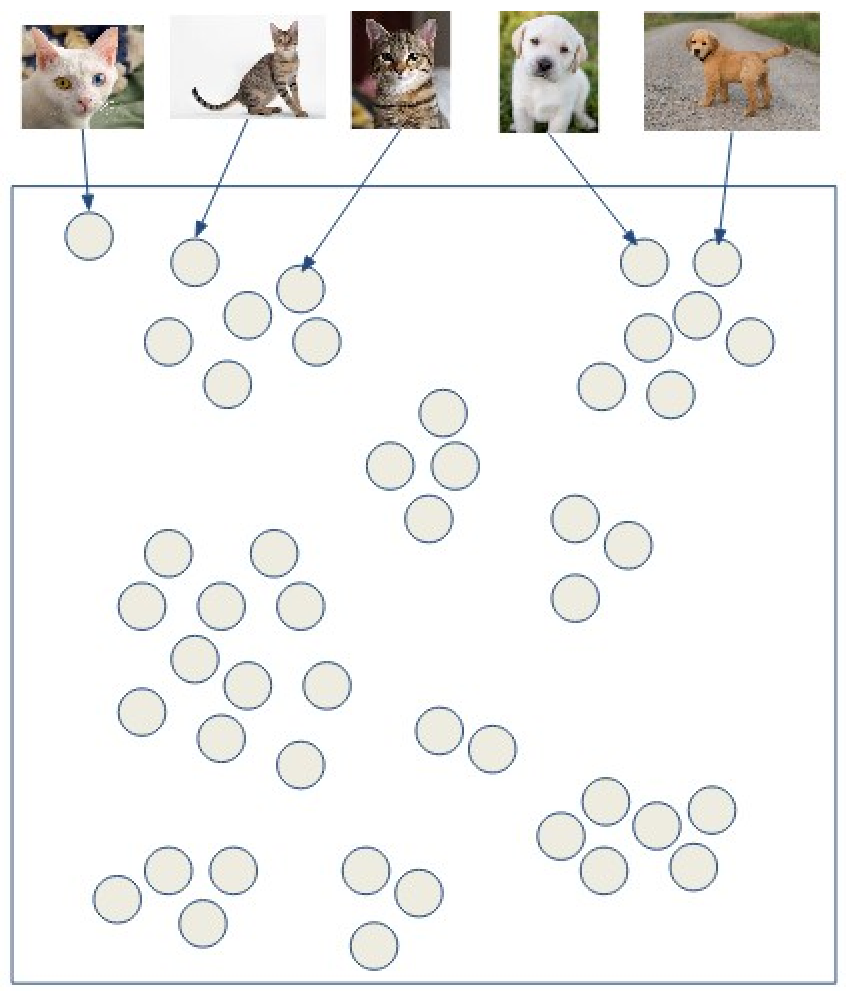
https://distill.pub

https://poloclub.github.io

https://karpathy.github.io

Explanation by design

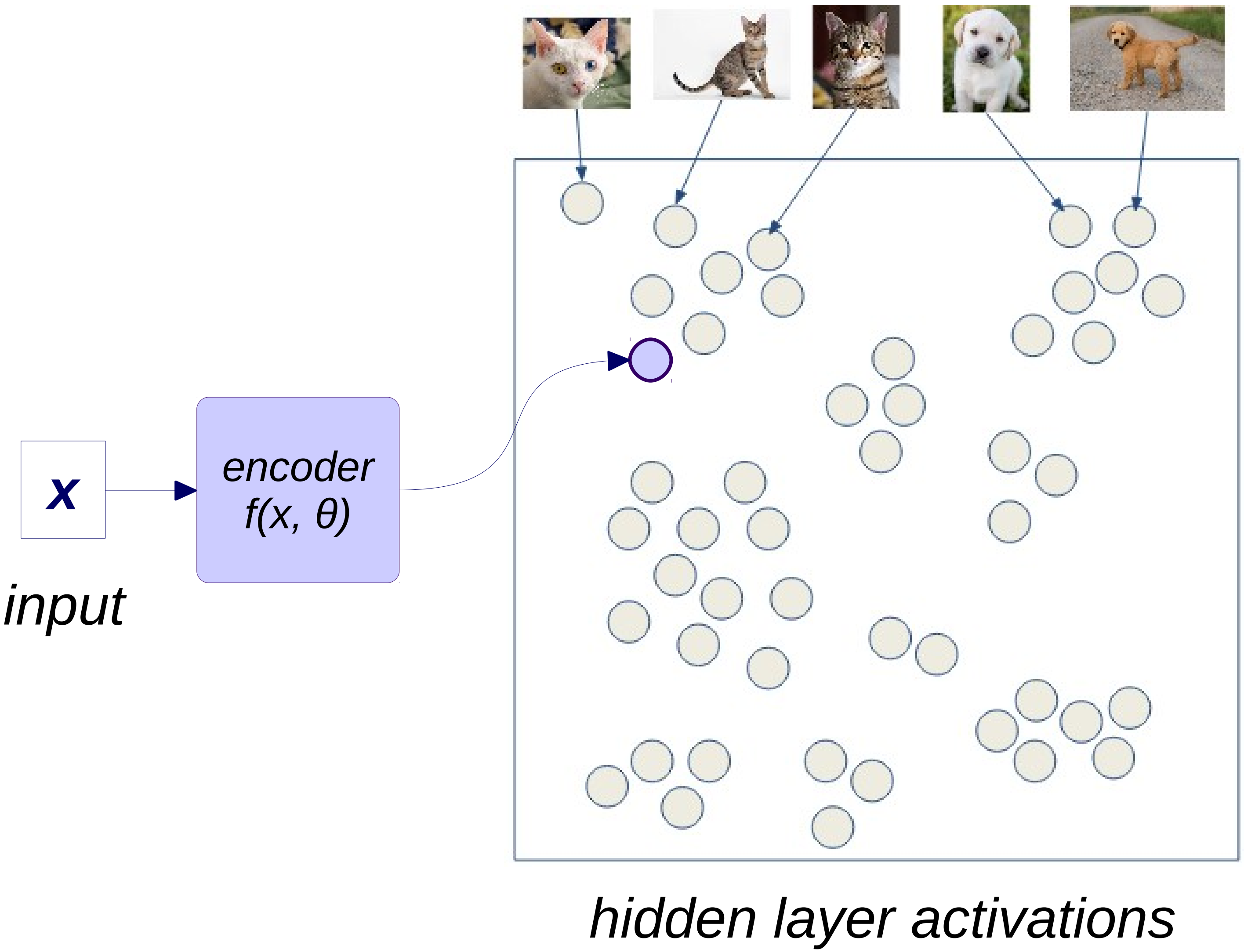
Idea: design architecture to be interpretable



hidden layer activations

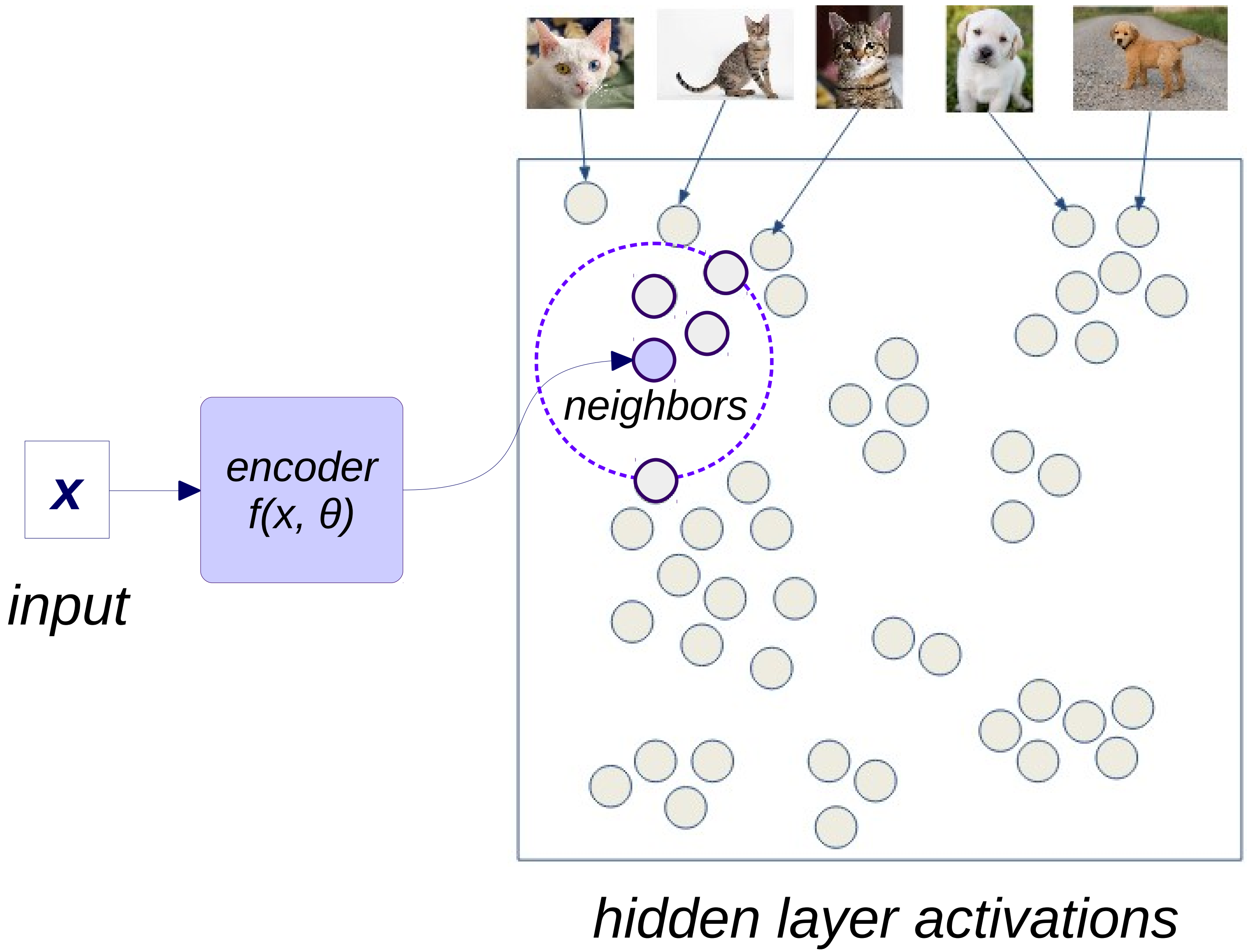
Explanation by design

Idea: design architecture to be interpretable



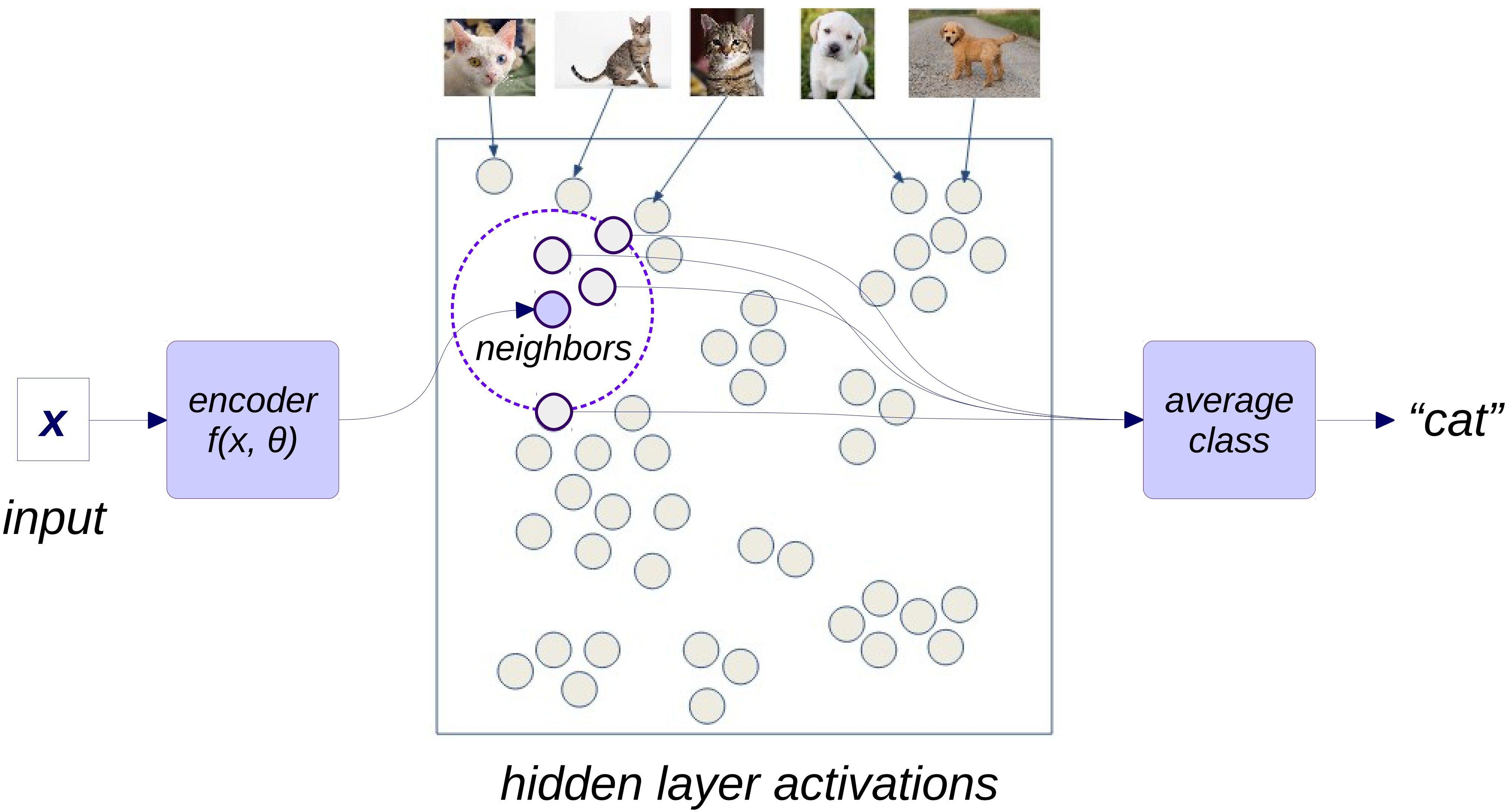
Explanation by design

Idea: design architecture to be interpretable



Explanation by design

Idea: design architecture to be interpretable



Explanation by design

Idea: design architecture to be interpretable

Prototype objects and answers: $\left(\hat{x}_{0,} \hat{y}_0\right), \ldots, \left(\hat{x}_N, \hat{y}_N\right)$

"Attention" weights: $$a\left(x, \hat{x}_i\right) = \frac{e^{\langle f(x, theta), f(\hat{x}_i, theta)\rangle}}{\sum_{j=0}^{N} e^{\langle f(x, theta), f(\hat{x}_j, theta)\rangle}}$$

Prediction by averaging: $$y^{pred}(x) = \sum_i \hat{y}_i \cdot a_i\left(x, \hat{x}_i\right)$$

Explanation by design

Idea: design architecture to be interpretable

Prototype objects and answers: $(\hat{x}_{0,} \hat{y}_0), \ldots, (\hat{x}_N, \hat{y}_N)$

"Attention" weights: $$a(x, \hat{x}_i) = \frac{e^{\langle f(x, theta), f(\hat{x}_i, theta)\rangle}}{\sum_{j=0}^{N} e^{\langle f(x, theta), f(\hat{x}_j, theta)\rangle}}$$

$$y^{pred}(x) = \sum_i \hat{y}_i \cdot a_i(x, \hat{x}_i)$$

Read more: KNN
arxiv.org/abs/1703.05175
arxiv.org/abs/1803.04765
arxiv.org/abs/1809.02847

Read more: Linear
arxiv.org/abs/1705.08078
arxiv.org/abs/1806.07538

The question of trust

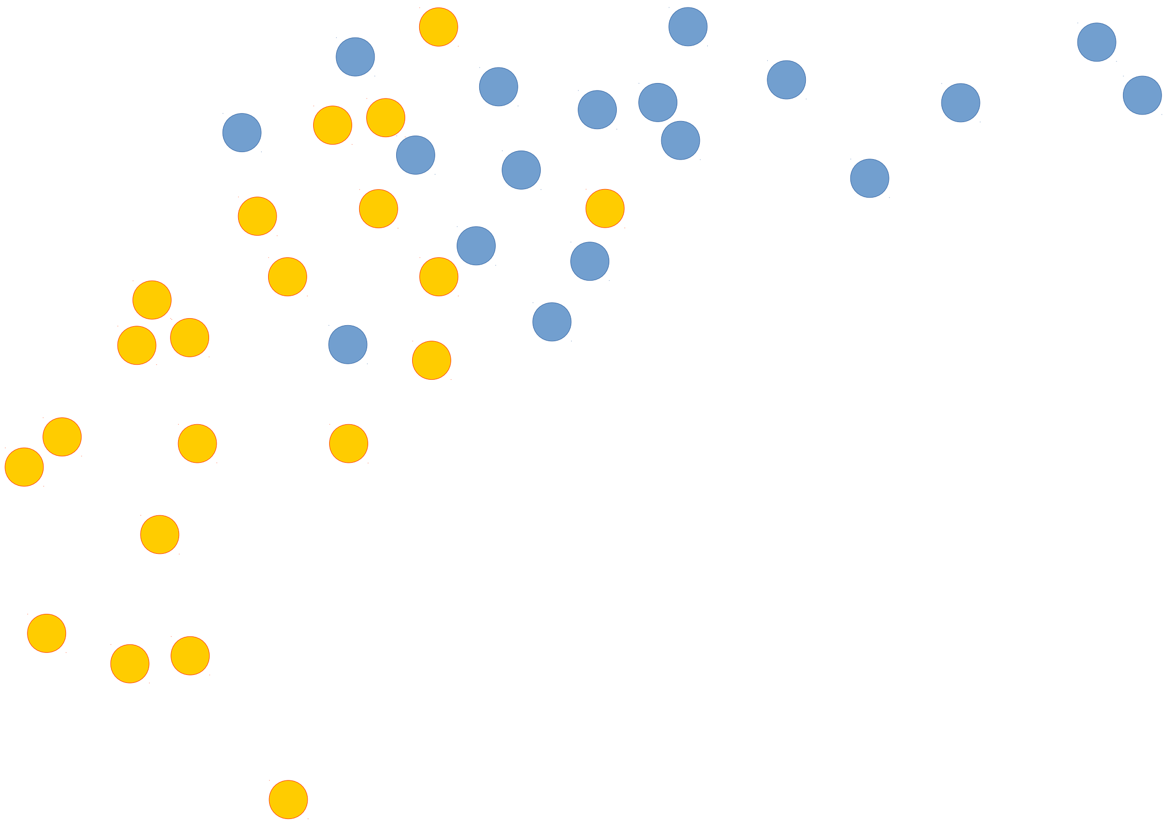
How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

Is my model certain about what it says?
Is there something wrong with this input?
Can I rely on this prediction?

Can I trust this data?
Is something missing?
Is there any bias?

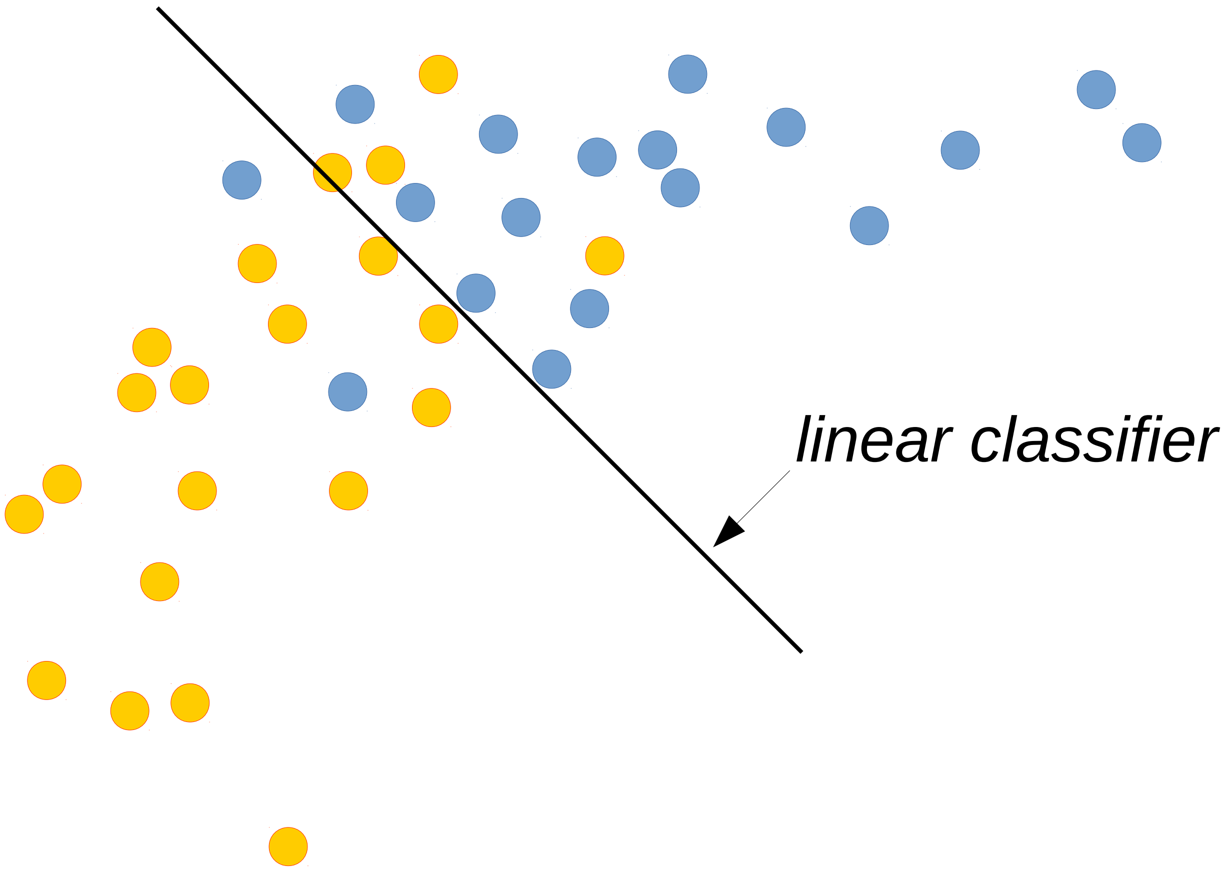
Types of uncertainty

example: binary classification



Types of uncertainty

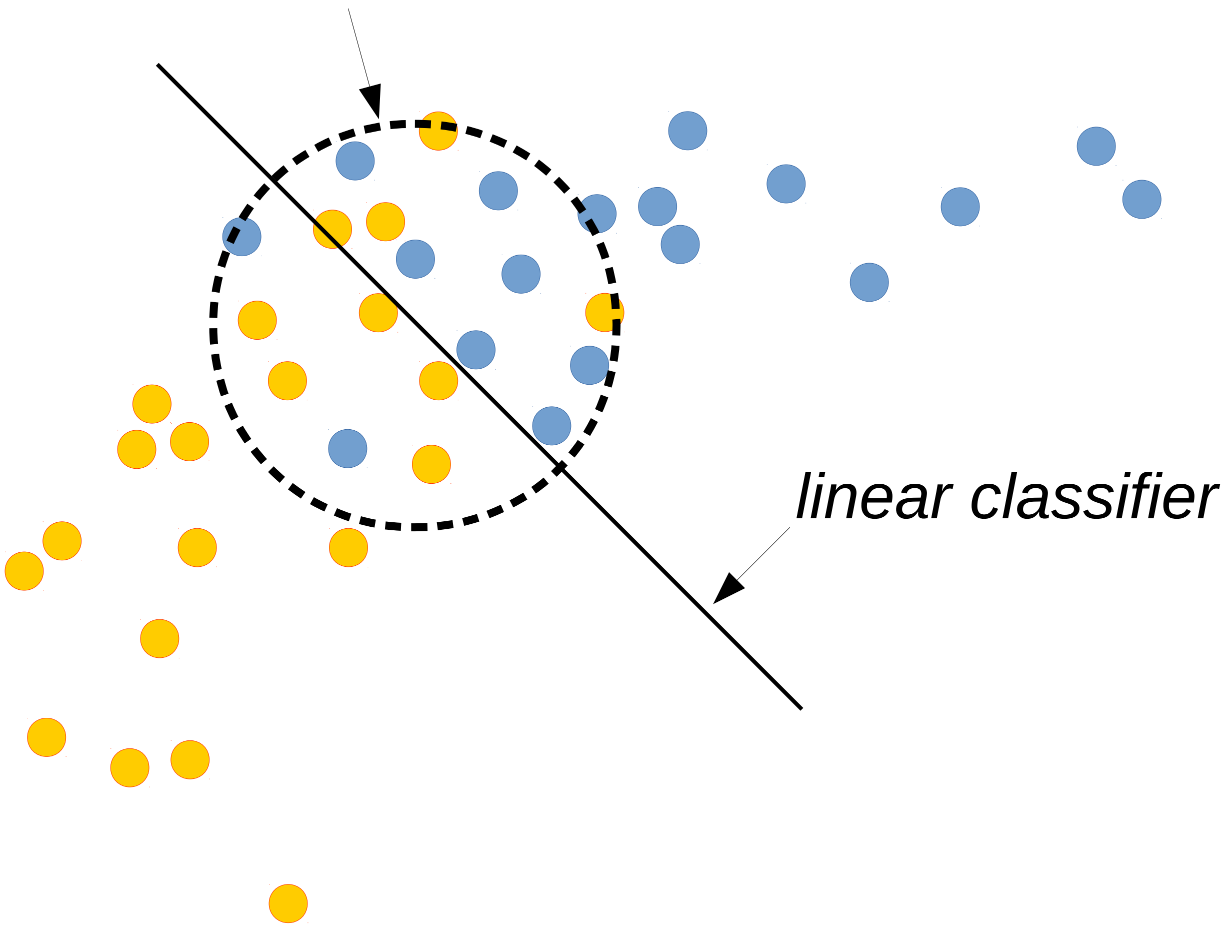
example: binary classification



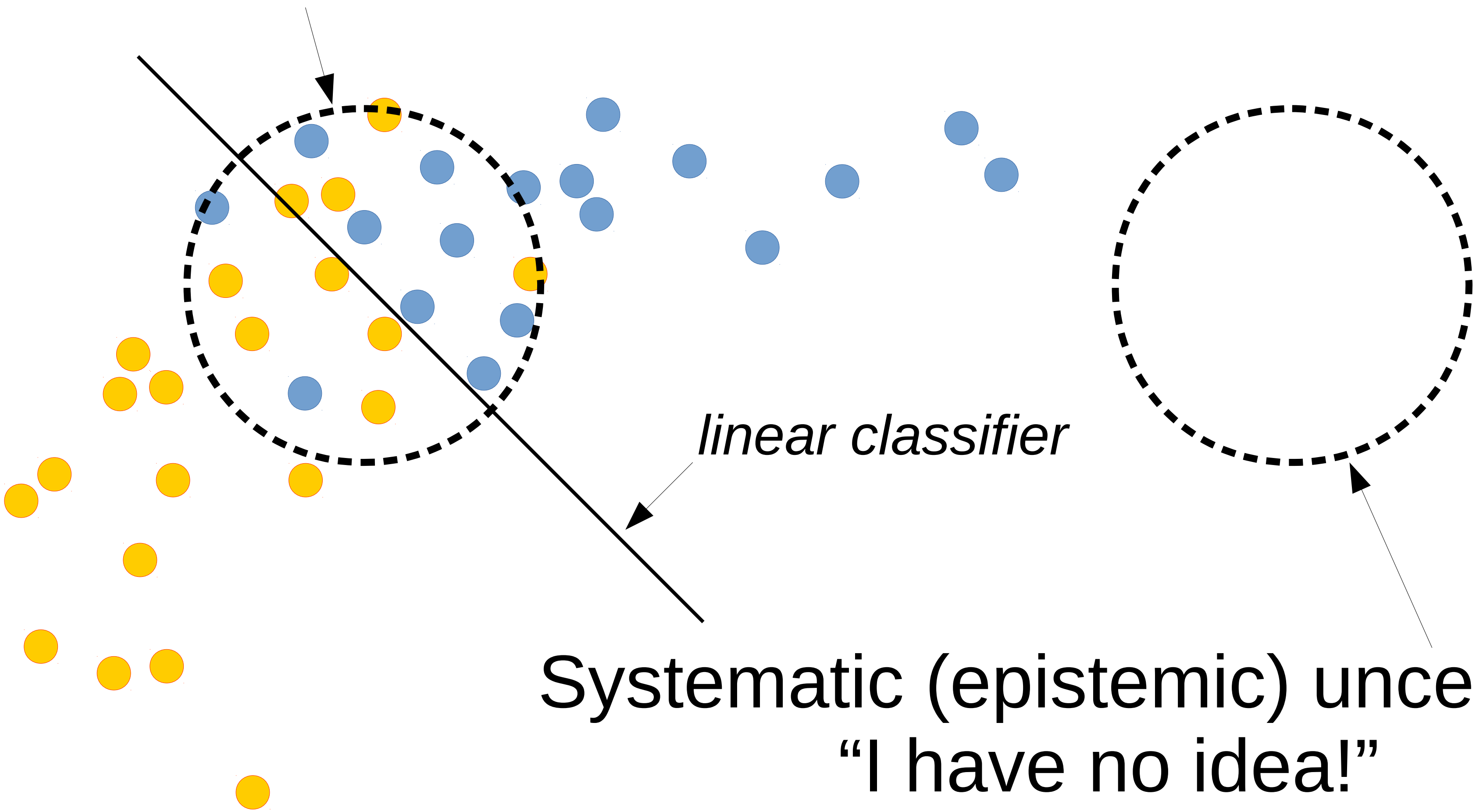
Types of uncertainty

Statistical (aleatoric) uncertainty

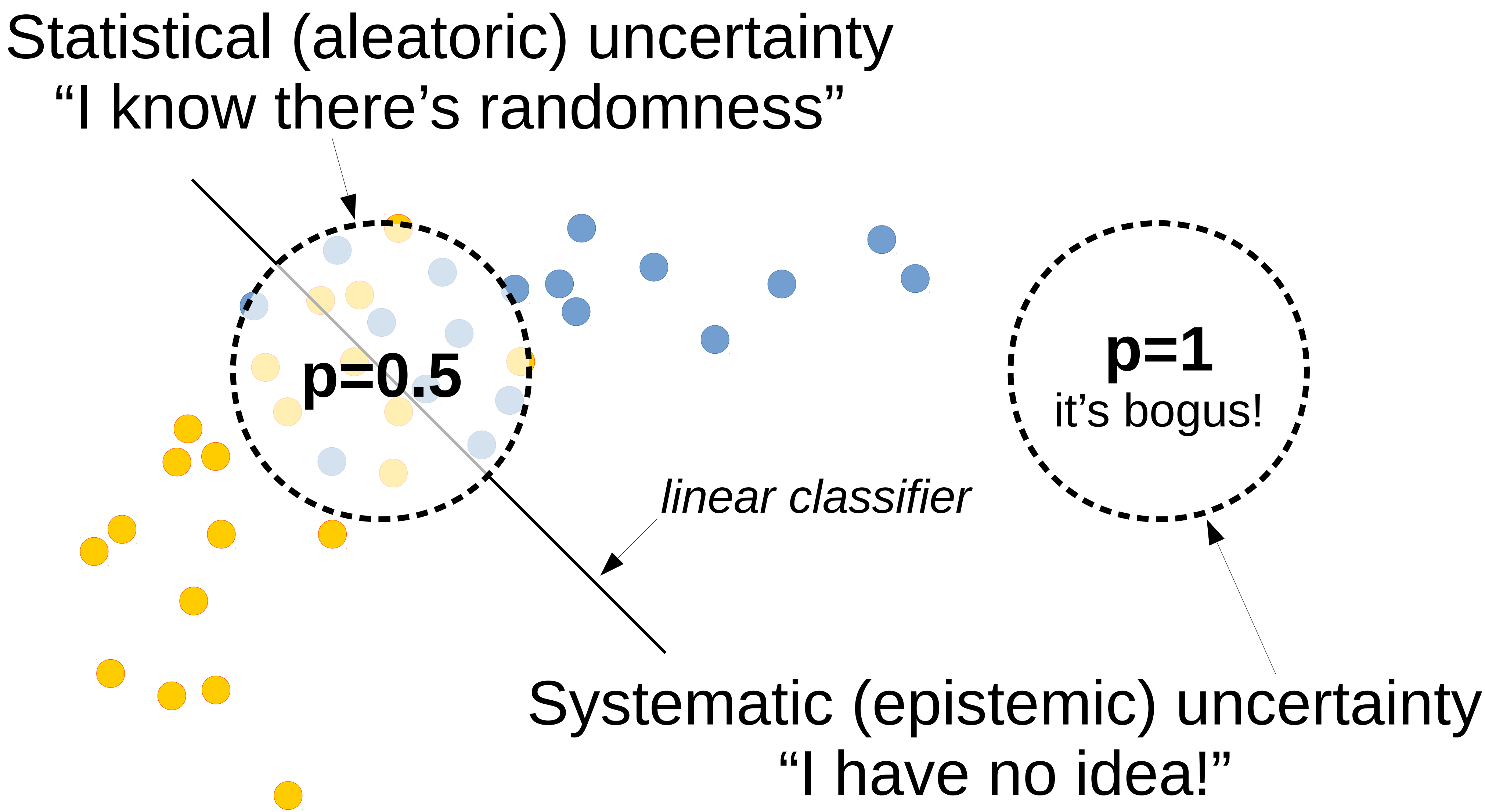
"I know there's randomness"



Types of uncertainty



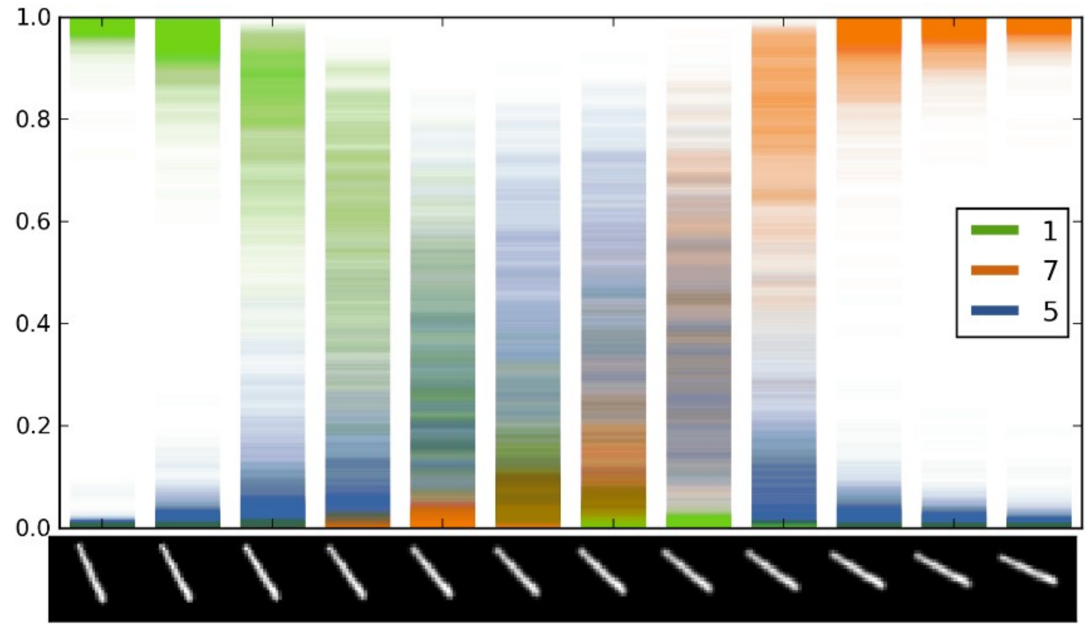
Types of uncertainty



Uncertainty from dropout

Idea:
measure how robust does your network perform under noise

Example (left):
use dropout and estimate variance



Uncertainty for different input images, source: arxiv.org/abs/1506.02142

Read more in the paper or in a blog post

Bayesian Neural Networks

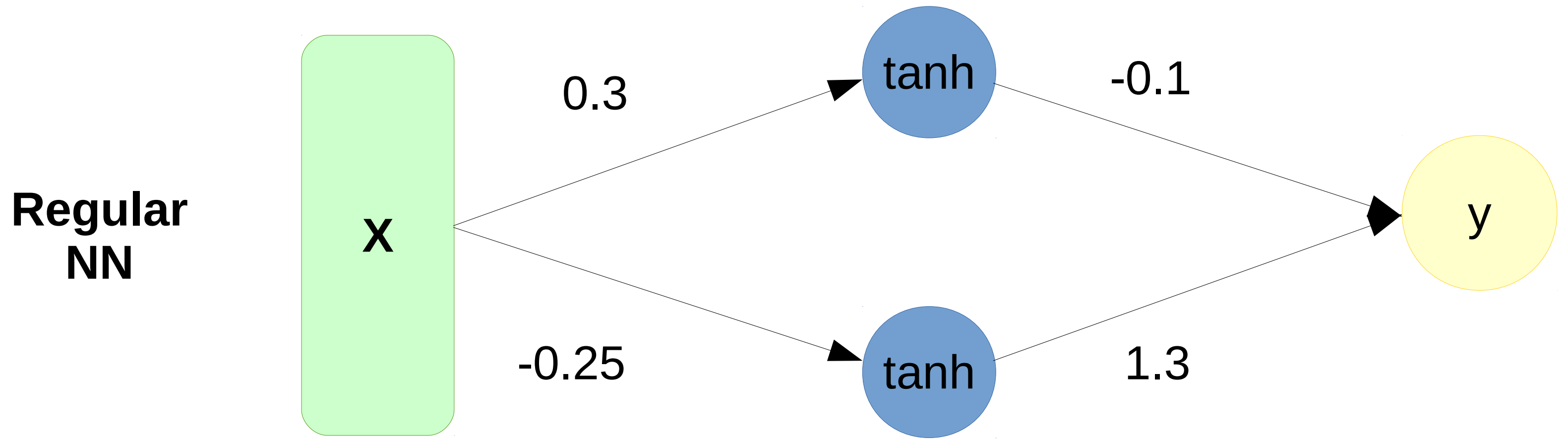
Disclaimer: this is a hacker's guide to BNNs!

It does not cover all the philosophy and general cases.

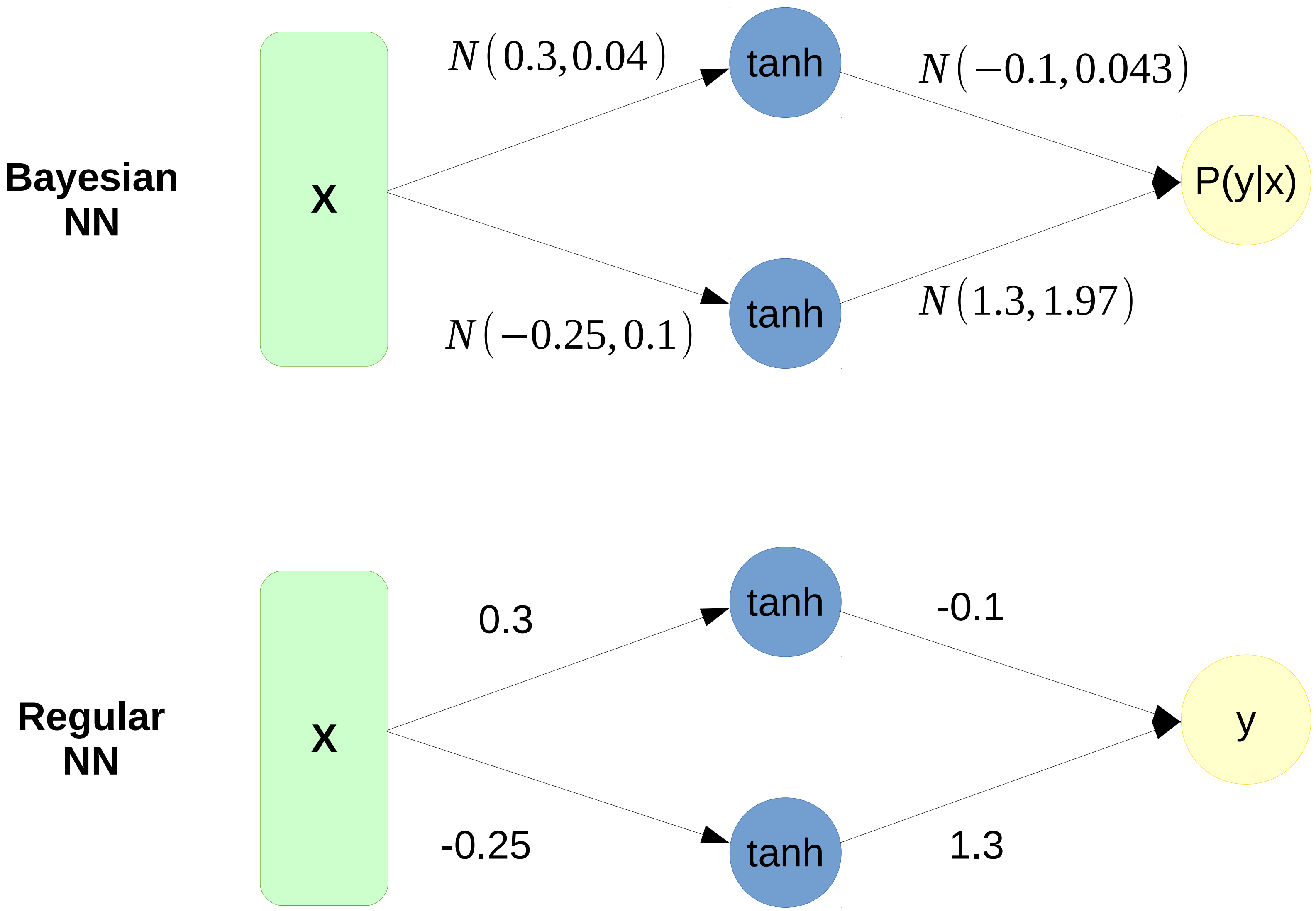
Bayesian Neural Networks

Disclaimer: this is a hacker's guide to BNNs!

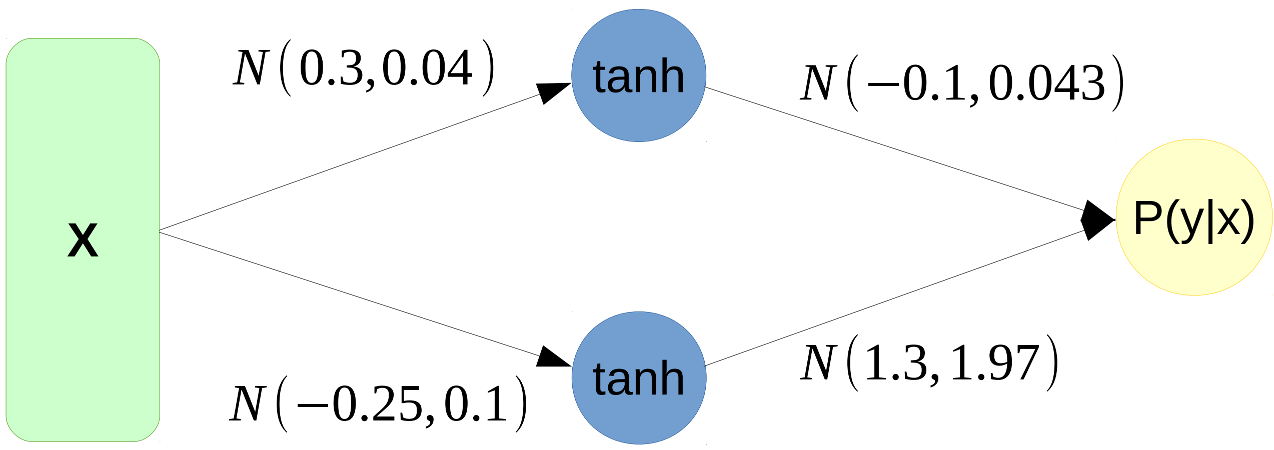
It does not cover all the philosophy and general cases.



Bayesian Neural Networks



Bayesian Neural Networks



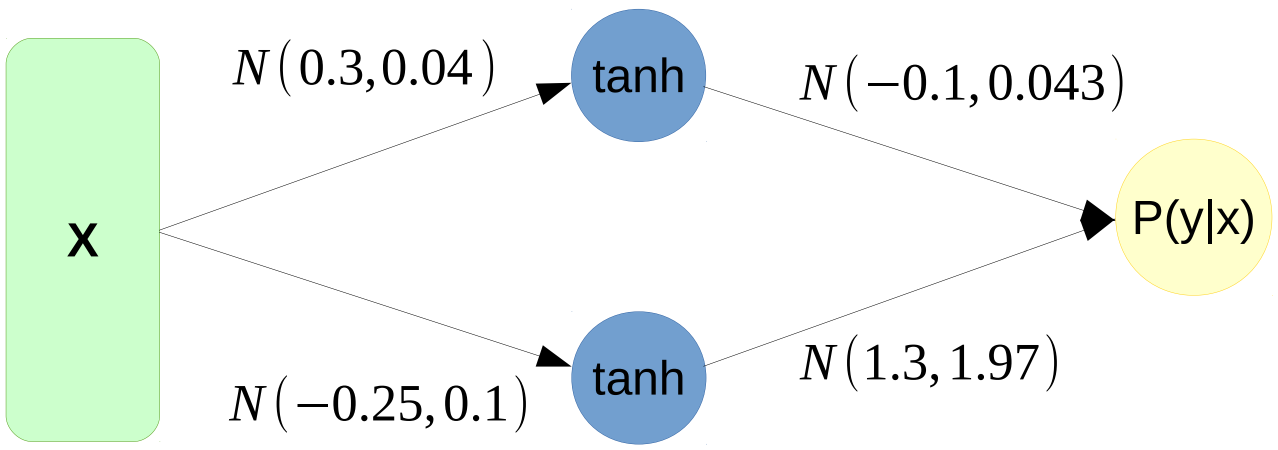
Idea:

- No explicit weights
  - Maintain parametric distribution on them instead!
    - Practical: fully-factorized normal or similar

$$q(\theta|\phi:[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta \sim q(\theta|\phi)} P(y|x,\theta)$$

Bayesian Neural Networks



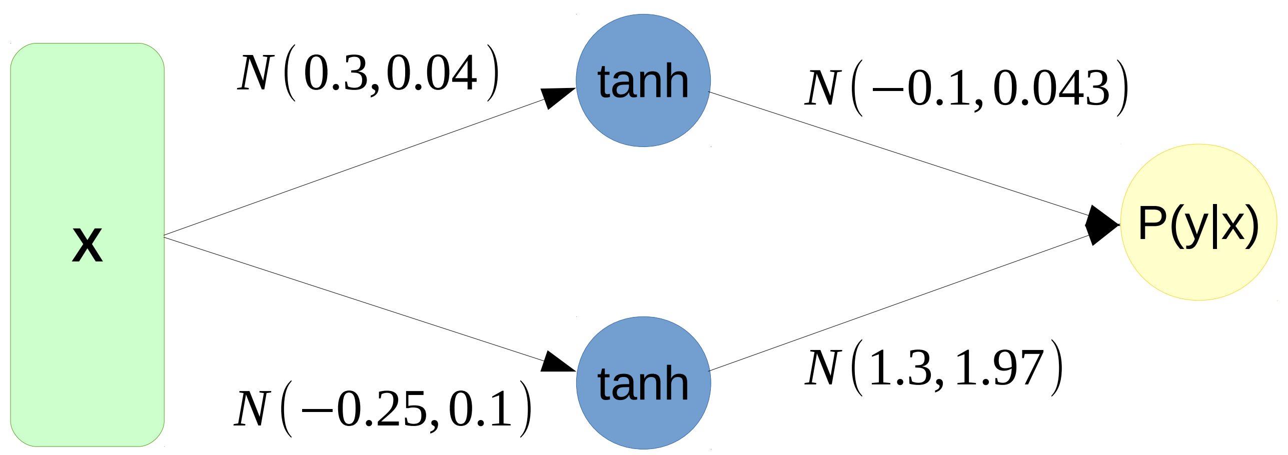
Idea:

- No explicit weights
  - Maintain parametric distribution on them instead!
    - Practical: fully-factorized normal or similar

$$q(\theta|\phi:[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta\sim q(\theta|\phi)} P(y|x,\theta)$$

Bayesian Neural Networks



Idea:

- No explicit weights
- Inference: sample from weight distributions, predict 1 “sample”
- To get distribution, aggregate K samples (e.g. with histogram)
  - Yes, it means running network **multiple times per one X**

$$P(y|x) = E_{\theta \sim q(\theta|\phi)} P(y|x,\theta)$$

Bayesian Neural Networks

Idea:

- No explicit weights
  - Maintain parametric distribution on them instead!
    - Practical: fully-factorized normal or similar

$$q(\theta|\phi:[\mu,\sigma]) = \prod_i N(\theta_i|\mu_i,\sigma_i)$$

$$P(y|x) = E_{\theta\sim q(\theta|\phi)} P(y|x,\theta)$$

- Learn parameters of that distribution (reparameterization trick)
  - Less variance: local reparameterization trick.

$$\mathring{\phi} = argmax_{\phi} E_{x_i, y_i \sim d} E_{\theta\sim q(\theta|\phi)} P(y_i|x_i,\theta)$$

wanna explicit formulae? *d = dataset*

Evidence Lower bound

d = dataset

$$-KL(q(\theta|\phi)\|p(\theta|d)) = -\int_\theta q(\theta|\phi)\cdot\log\frac{q(\theta|\phi)}{p(\theta|d)}$$

$$-\int_\theta q(\theta|\phi)\cdot\log\frac{q(\theta|\phi)}{\left[\frac{p(d|\theta)\cdot p(\theta)}{p(d)}\right]} = -\int_\theta q(\theta|\phi)\cdot\log\frac{q(\theta|\phi)\cdot p(d)}{p(d|\theta)\cdot p(\theta)}$$

$$-\int_\theta q(\theta|\phi)\cdot\left[\log\frac{q(\theta|\phi)}{p(\theta)} - \log p(d|\theta) + \log p(d)\right]$$

$$[E_{\theta\sim q(\theta|\phi)}\log p(d|\theta)] - KL(q(\theta|\phi)\|p(\theta)) + \log p(d)$$

loglikelihood **-distance to prior** **+const**

Evidence Lower bound

$$\phi = \underset{\phi}{argmax}\left(-KL\left(q\left(\theta|\phi\right) \| p\left(\theta|d\right)\right)\right)$$

$$\underset{\phi}{argmax}\left(\underbrace{\left[E_{\theta \sim q(\theta|\phi)} \log p\left(d|\theta\right)\right]}_{\text{fit to the data}} - KL\underbrace{\left(q\left(\theta|\phi\right) \| p\left(\theta\right)\right)}_{\text{don't be too certain}}\right)$$

Evidence Lower bound

$$\phi = \underset{\phi}{argmax}\left(-KL\left(q\left(\theta|\phi\right)||p\left(\theta|d\right)\right)\right)$$

$$\underset{\phi}{argmax}\left(\left[E_{\theta\sim q\left(\theta|\phi\right)}\log p\left(d|\theta\right)\right]-KL\left(q\left(\theta|\phi\right)||p\left(\theta\right)\right)\right)$$

Can we perform gradient ascent directly?

Reparameterization trick

$$\phi = \underset{\phi}{argmax}\left(-KL\left(q(\theta|\phi)\,\|\,p(\theta|d)\right)\right)$$

$$\underset{\phi}{argmax}\left(\underbrace{\left[E_{\theta\sim q(\theta|\phi)}\log p(d|\theta)\right]}_{\text{Use reparameterization trick}} - \underbrace{KL\left(q(\theta|\phi)\,\|\,p(\theta)\right)}_{\text{simple formula (for normal q)}}\right)$$

BNN likelihood

What does this log P(d|...) mean?

$$E_{\theta\sim N(\theta|\mu_\phi,\sigma_\phi)}\log p(d|\theta) = E_{\psi\sim N(0,1)}\log p\left(d\,|\left(\mu_\phi+\sigma_\phi\cdot\psi\right)\right)$$

Reparameterization trick

$$\phi = \underset{\phi}{argmax}\left(-KL\left(q\left(\theta|\phi\right)\|p\left(\theta|d\right)\right)\right)$$

$$\underset{\phi}{argmax}\left(\left[E_{\theta\sim q\left(\theta|\phi\right)}\log p\left(d|\theta\right)\right]-KL\left(q\left(\theta|\phi\right)\|p\left(\theta\right)\right)\right)$$

In other words,
$\Sigma_{x,y\sim d}$ log p(y|x,μ+σψ)

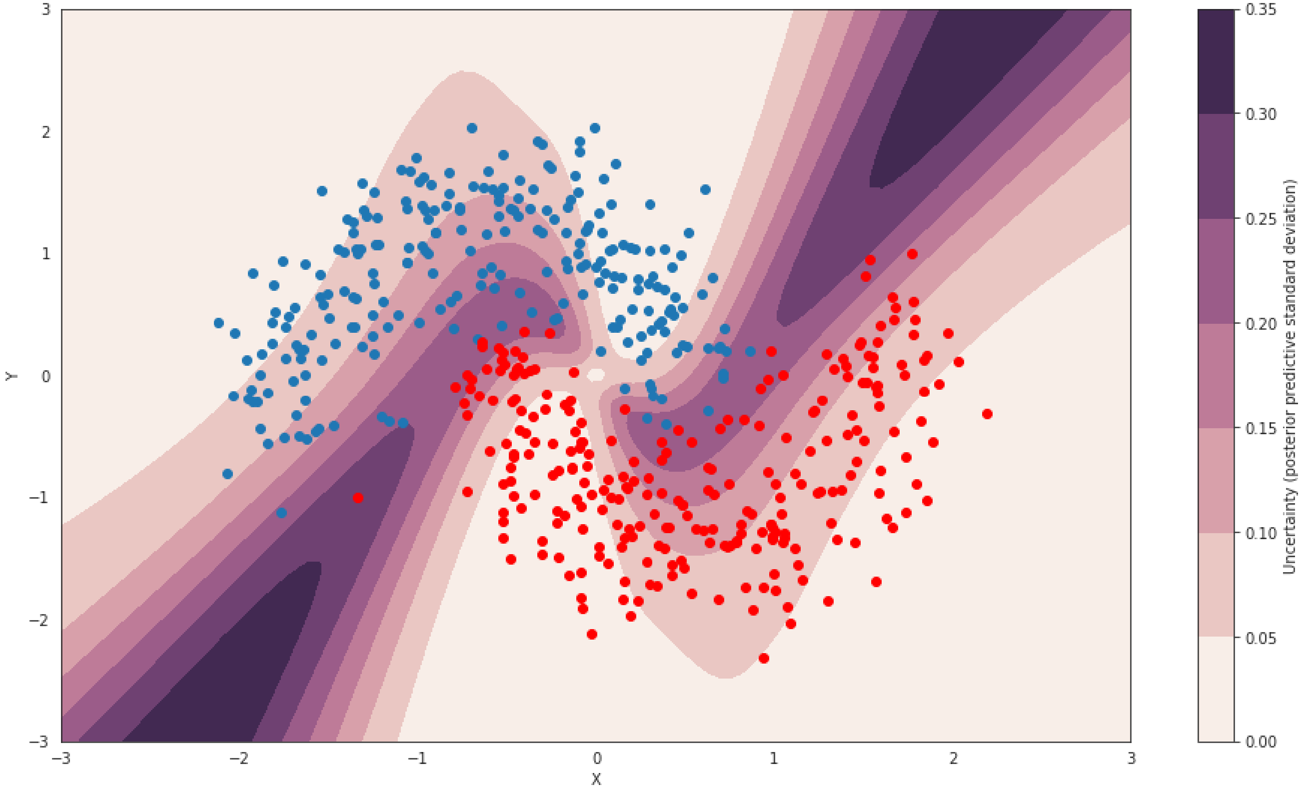
BNN likelihood

$$E_{\theta\sim N\left(\theta|\mu_\phi,\sigma_\phi\right)}\log p\left(d|\theta\right)=E_{\psi\sim N\left(0,1\right)}\log p\left(d|\left(\mu_\phi+\sigma_\phi\cdot\psi\right)\right)$$

Bayesian Neural Networks

Estimating uncertainty:

1. sample weights several times
2. predict by averaging outputs
3. uncertainty = standard deviation



Read more...

Papers on uncertainty

bayesian neural networks: blog post
prior networks: arxiv.org/abs/1802.10501
batchnorm: arxiv.org/abs/1802.04893
dropout: arxiv.org/abs/1506.02142
video stuff: youtube.com/watch?v=HRfDiqgh6CE

The question of trust

How can I explain my model's prediction?
Why did it make this decision/mistake?
What features does it rely on?

Is my model certain about what it says?
Is there something wrong with this input?
Can I rely on this prediction?

Can I trust this data?
Is something missing?
Is there any bias?

Exploratory data analysis

There should be some TSNE slides
but there won't cuz you already know TSNE

Exploratory data analysis

How many dimensions can you show on a plot?

Exploratory data analysis

Physics data / images / sound = high dimensional

Dimensionality reduction: PCA

Show the fish :)

Manifold learning

Idea: learn representations so that ...

Multidimensional scaling

Example

Stochastic Neighborhood Embedding

Example, add distill.pub url

TSNE

Example, add distill.pub url

TSNE + deep encoder

Example, add distill.pub post

Thank you

Outro text