

Beyond prediction

Explainability and confidence

Let's talk about last lab a bit

The image feature actually helps with some tweaks

- Remove weak features

- Do it by price section

- Treat different types differently

“[...] For a bank to predict people who would default we build two models: One that was very simple, is actually linear prediction of some kind or a local linear prediction [...] that economist could interpret [...], and then we built the second model, big neural net with some sort of latent variables [...] very sophisticated, worked very well, way better than the other one.

When presented to them, **“which one you want, the one that is explainable, or the one that actually works”**

Every single time I've talked to people who wanted to use a machine learning system, they look at you in the eye and say “you know, is very important to have an explanation that is used for making decisions, [...] we aren't interested in anything unless you have an explanation[...]

And then you show them two models: one that has an explanation (kind of) [...]



Yann LeCun
The Great AI Debate - NIPS2017

“Every single time they'll take the second one! Which means they don't really care about explainability, it's a way for them to be reassured, and most people are more reassured big a good AUC curve than the possible explanations of the decision a model internally makes.

Also NNnets are not black boxes, you can do sensitivity analysis to evaluate how each value impacts the score”

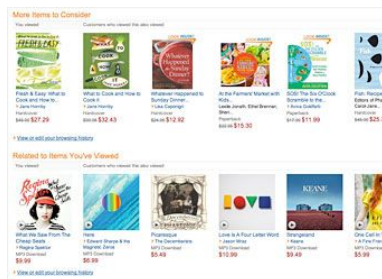
ML products

Customer facing

Recommendation systems,
maps (traffic estimate),
speech2text

Best guess by the model

Mostly automatic (check deposit
by app)



Internal facing

Loan applications, demand
forecasting

Can say “I’m not sure”

Human in the loop

Machine-assisted

Need to say why or give
confidence level



Two levels of understanding

Model level

- Describes the model tendency

- Talks about the behavior on training data

Output level

- Attributes model decision for a given test sample to different features

Model level: Simple example

Logistic regression

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i,$$

You can say which feature is important from size of the coefficients.

Pros: Simple, easy to understand

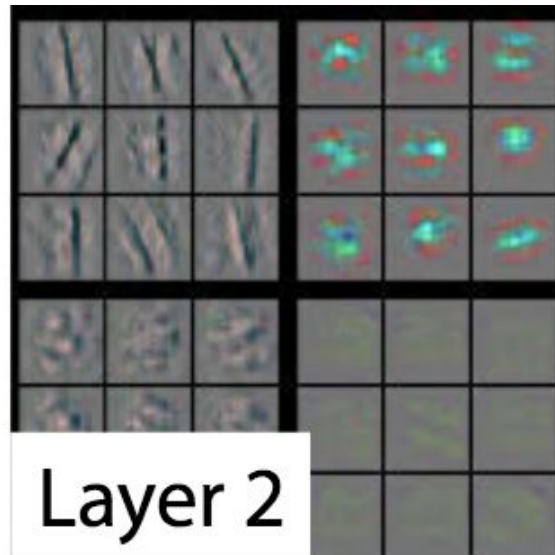
Cons: Only model linear effects

Model level: visualizing filters

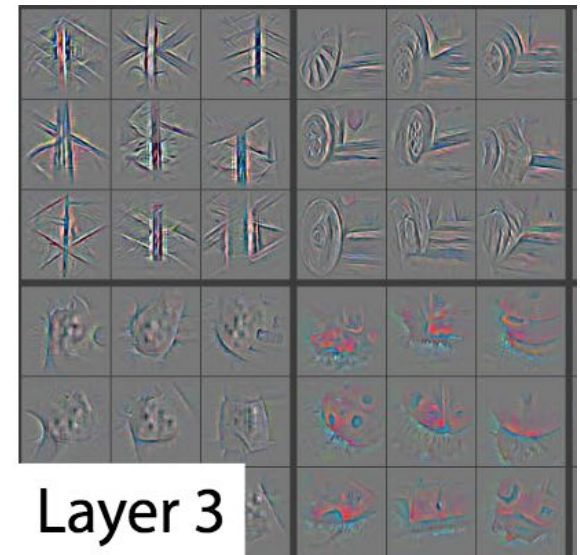
Plot out the weights of the CNN filters...not so useful in practice



Layer 1



Layer 2



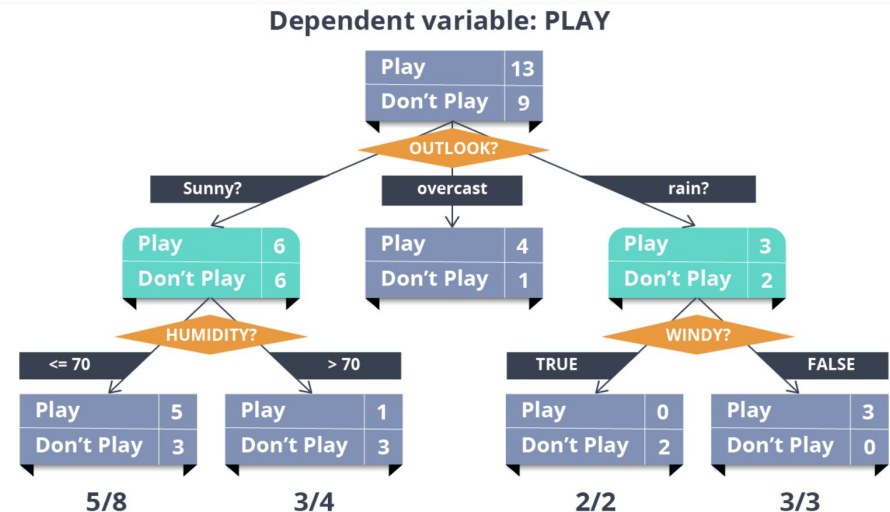
Layer 3

Model level: feature importance in random forests

Assign scores based on how the features are use

A node that splits better (better purity at children): high score

A node with more training samples: high weight



Usefulness of model level

Feature selection

Gives you confidence that the model is learning reasonable things

Two levels of understanding

Model level

- Describes the model tendency

- Talks about the behavior on training data

Output level

- Attributes model decision for a given test sample to different features

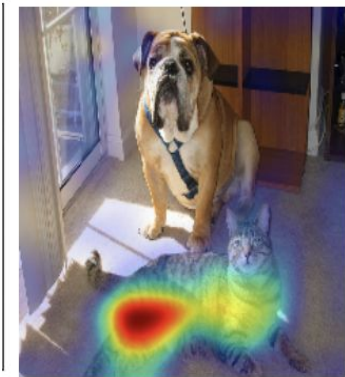
Output level: key ideas

A feature is important if I tweak the feature and the output change a lot.

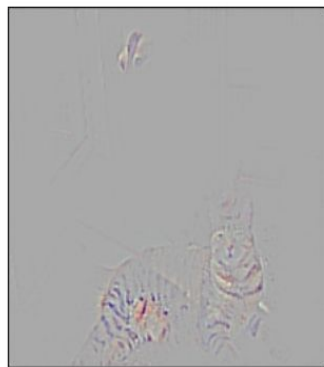
If I tweak the output, how does the gradient flows

Gradient descent onto the features!

Output level: Gradient-weighted Class Activation Mapping (Grad-CAM)



(c) Grad-CAM 'Cat'



(d) Guided Grad-CAM 'Cat'



(i) Grad-CAM 'Dog'



(j) Guided Grad-CAM 'Dog'



Ground truth: volcano



Predicted: sandbar

(a)



Ground truth: volcano

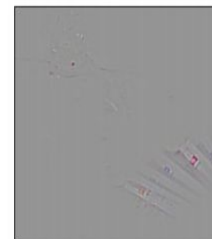


Predicted: car mirror

(b)

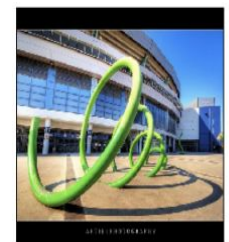


Ground truth: beaker



Predicted: syringe

(c)



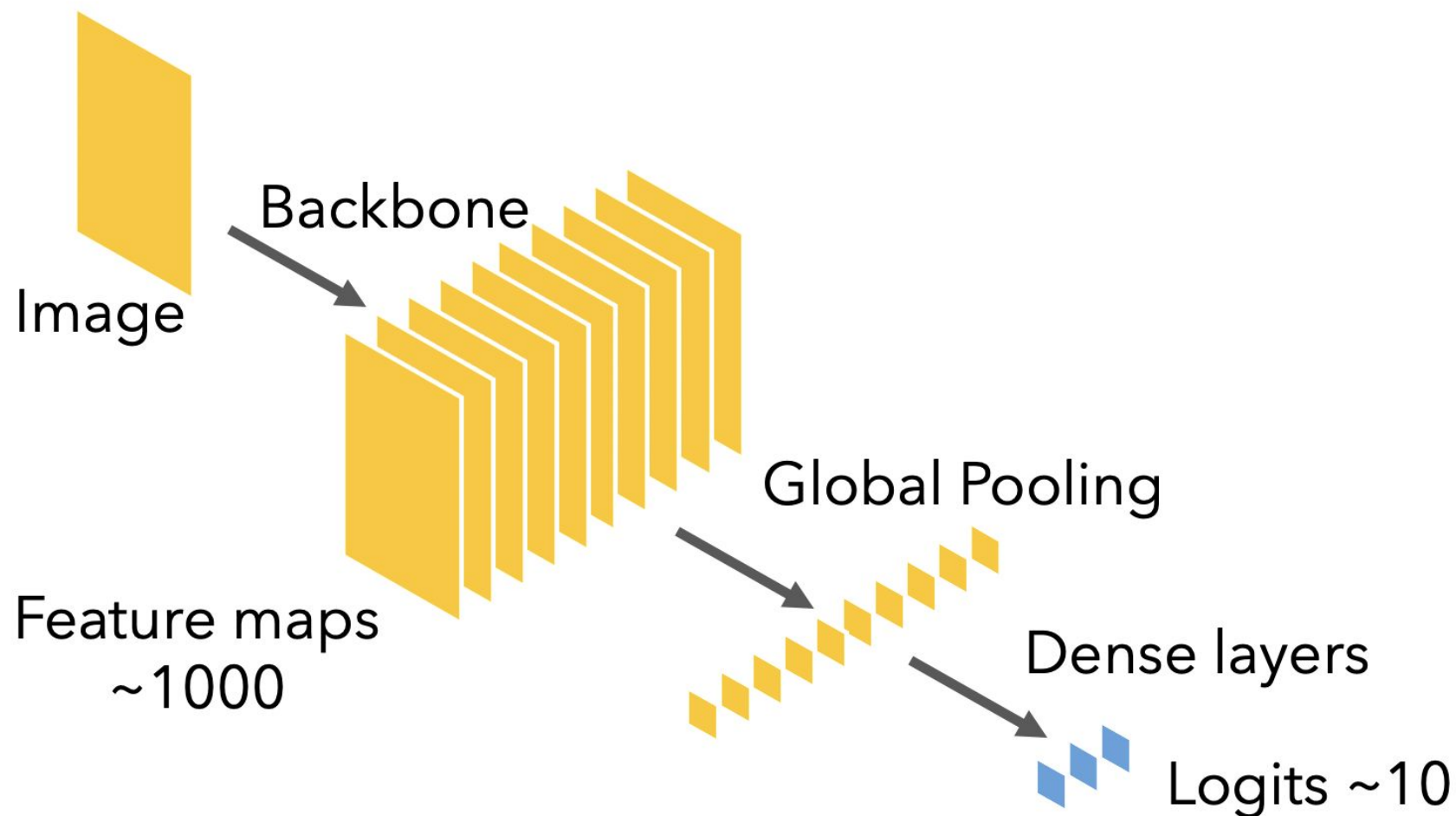
Ground truth: coil



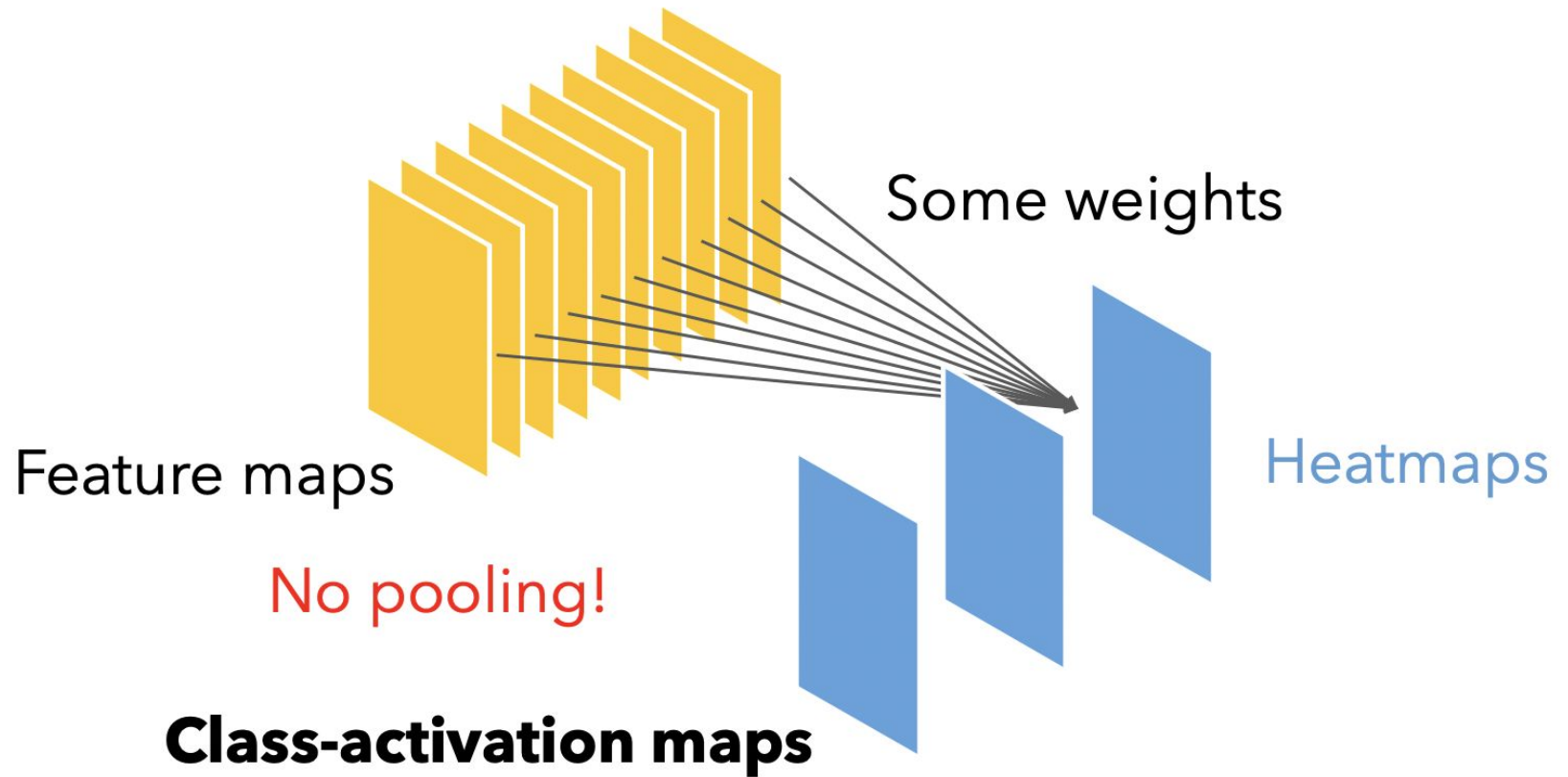
Predicted: vine snake

(d)

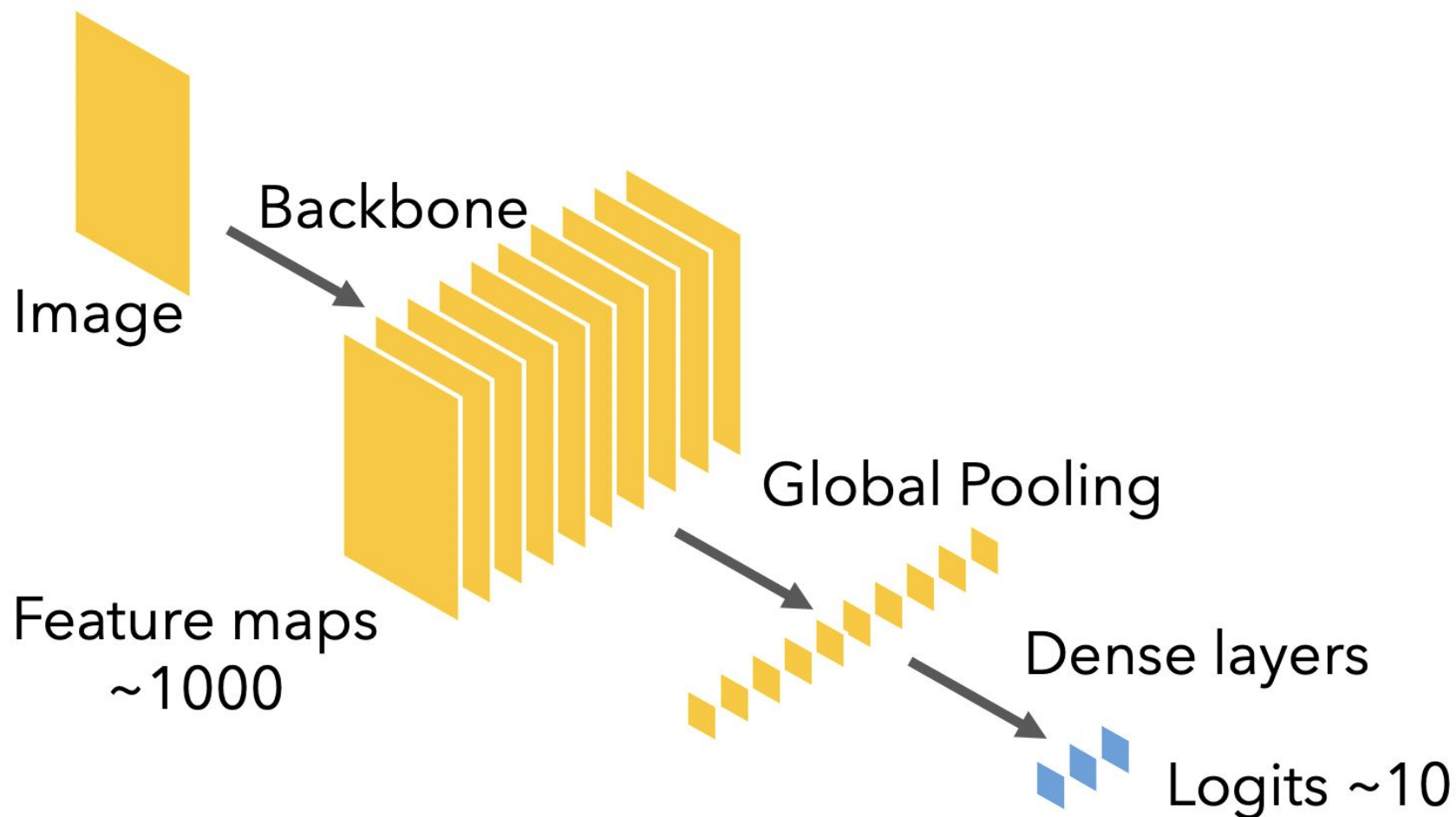
A simple CNN classifier



CAM



Grad-CAM in action



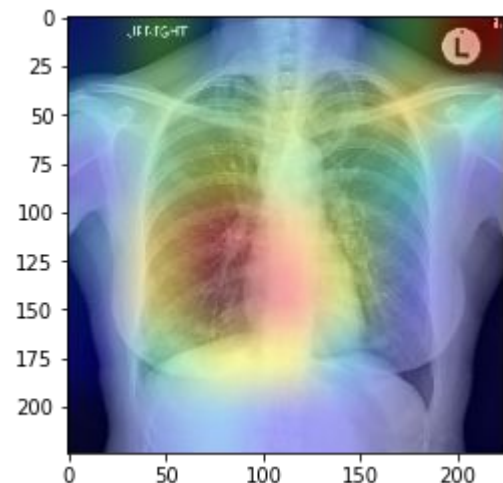
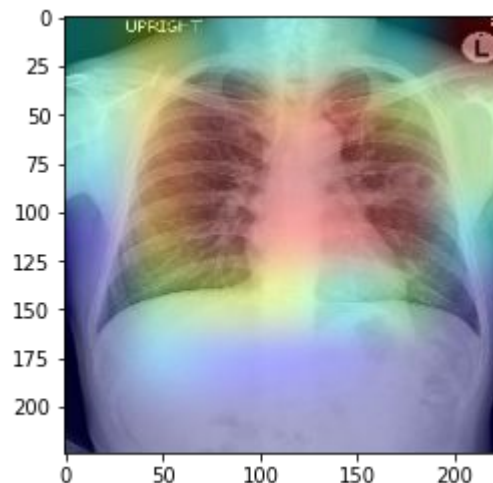
Output level: key ideas

A feature is important if I tweak the feature and the output change a lot.

Mostly useful for images types

Have a simpler model (**surrogate model**) explains the complicated model.

Converting things to logistic regression



Additive feature attribution

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i,$$

Simplified input features z' have binary values

z' can recover original features x , $x = h_x(z')$ (This mapping depends on x)

Goal: make $g(z') = f(h_x(z'))$. Then we can explain f in terms of simplified features. (**Solved by optimization**)

Example of simplified inputs

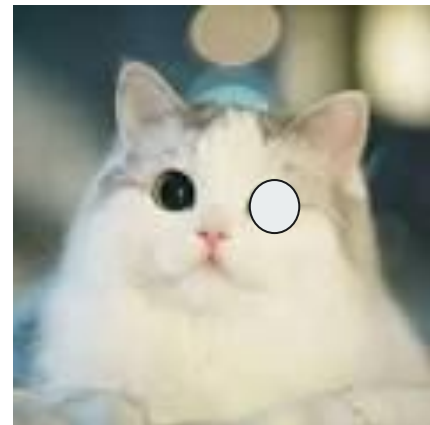
Original x = image

$z = 0$ if patch is not present

$z = 1$ if patch is present

$h_x(z) = x$ if, $z=1$

$h_x(z) = x$ with missing patch, if $z=0$



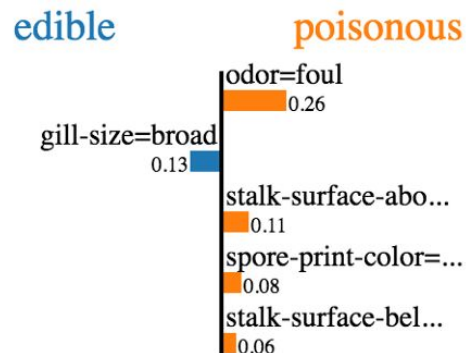
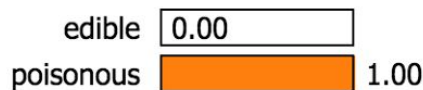
Additive feature attribution

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i,$$

Many methods can be considered as additive feature attribution. For example LIME and SHAP

LIME

Prediction probabilities



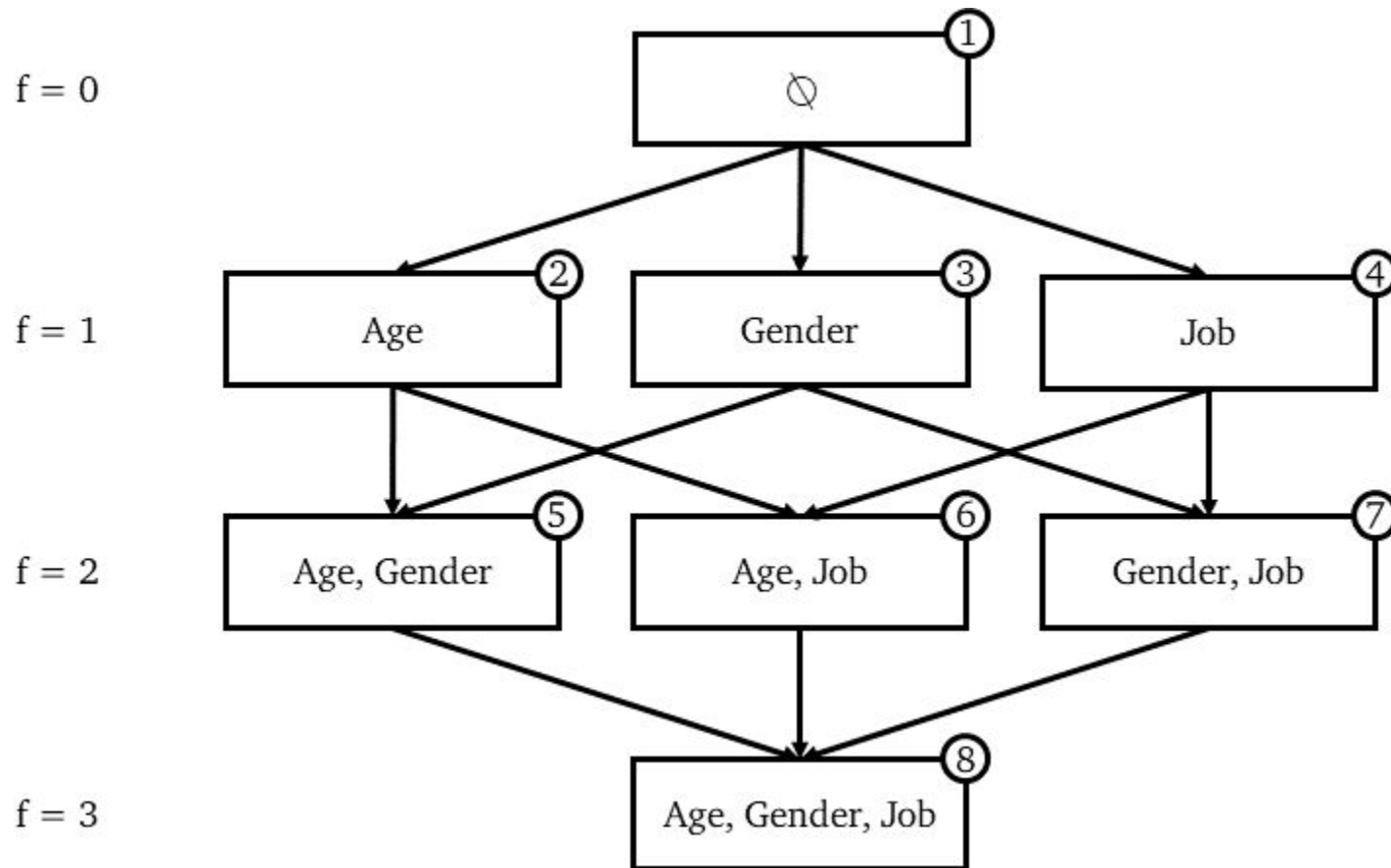
Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

SHAP

Based on the concept of shapely values
“Expected contribution” of each feature

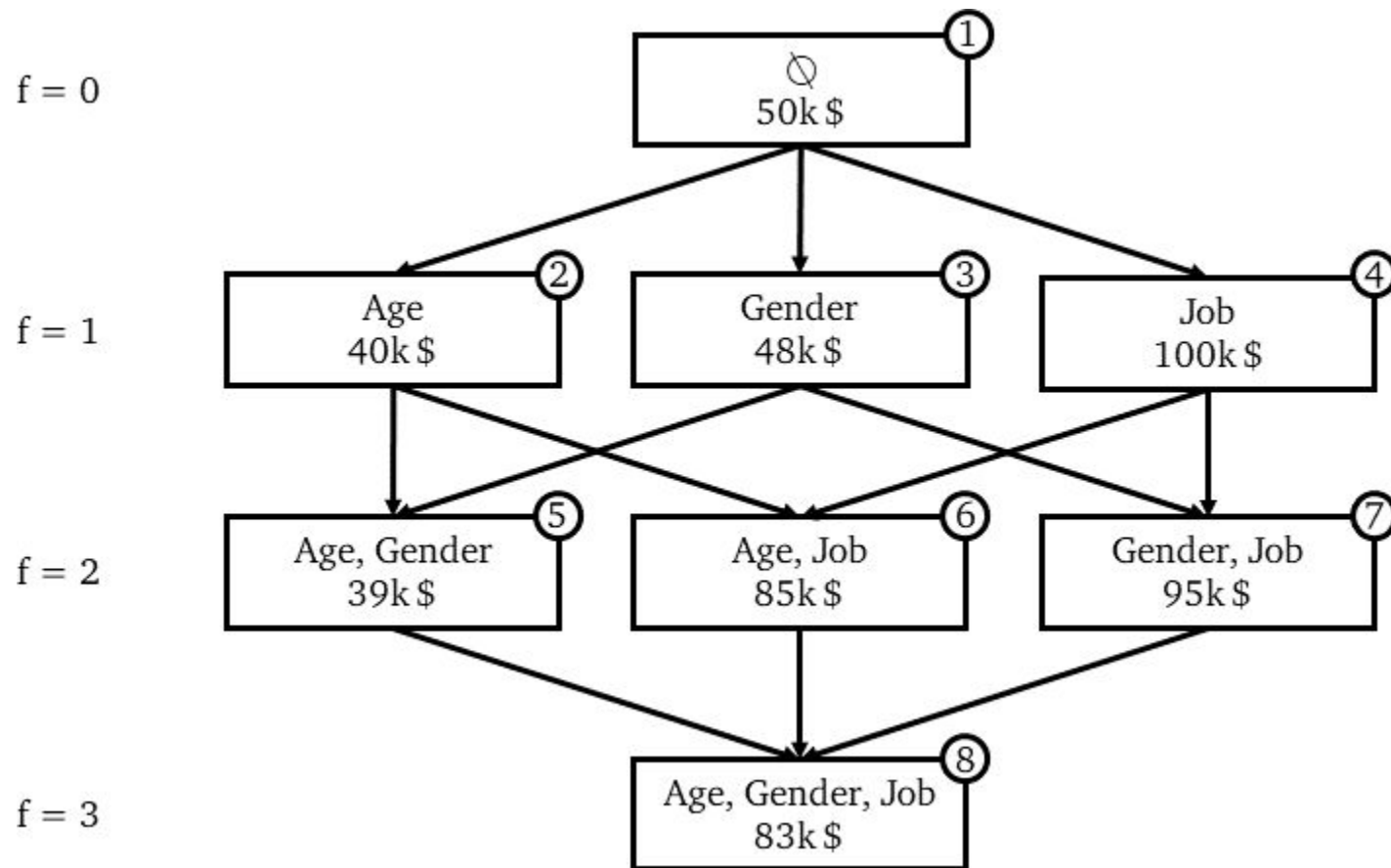
Shapely values

Suppose we have 3 features
Age, Gender, Job
Predicting income



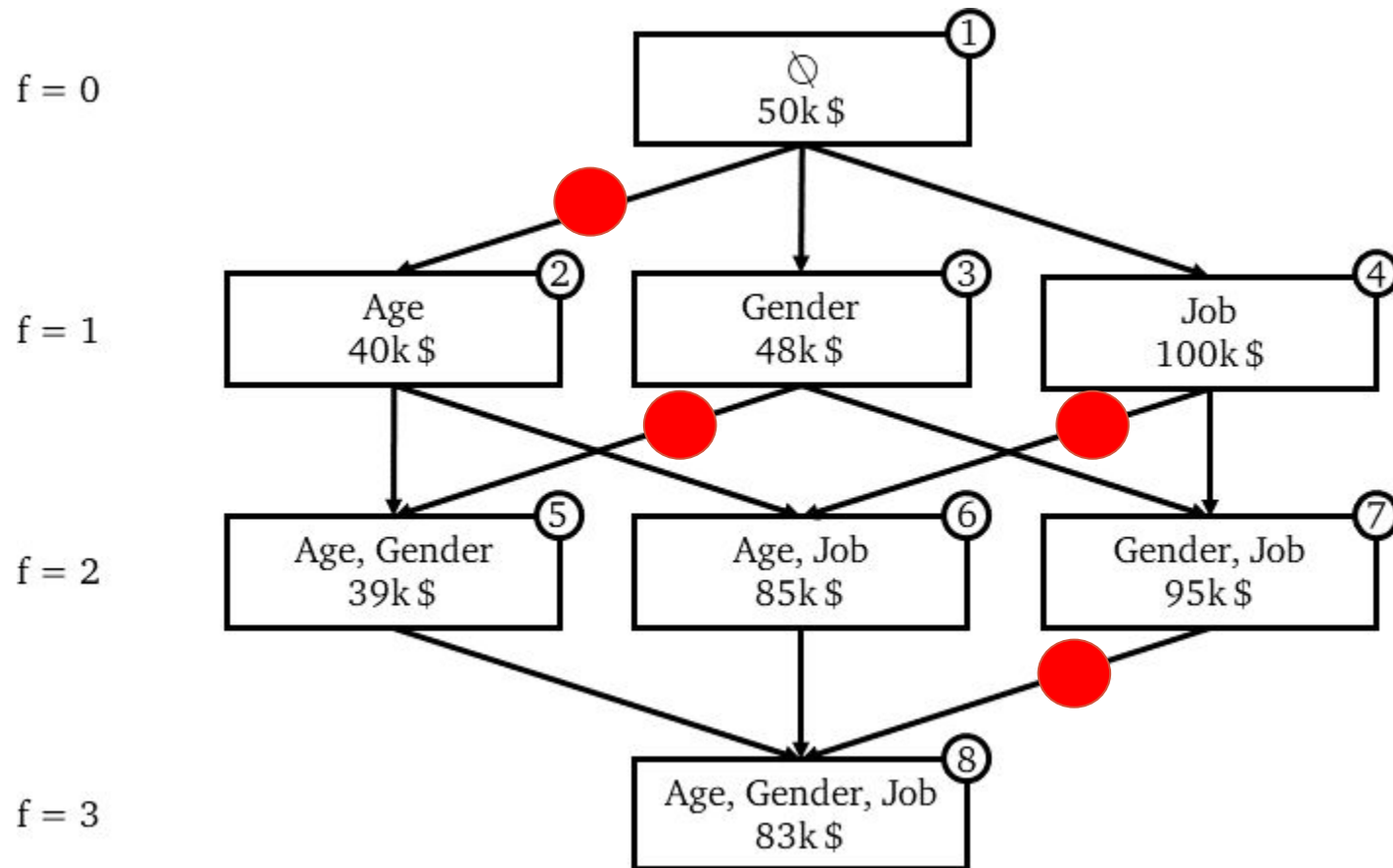
Shapely values

For a given data sample we can predict



Shapely values

The contribution of age feature is the “average” of all possible addition of the age feature.



SHAP

Based on the concept of shapely values

“Expected contribution” of each feature

SHAP approximates shapely values by sampling

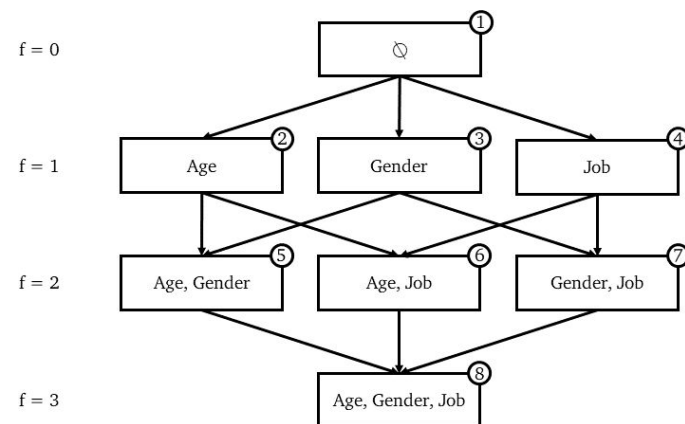
KernelExplainer

Sample subset of features to compute

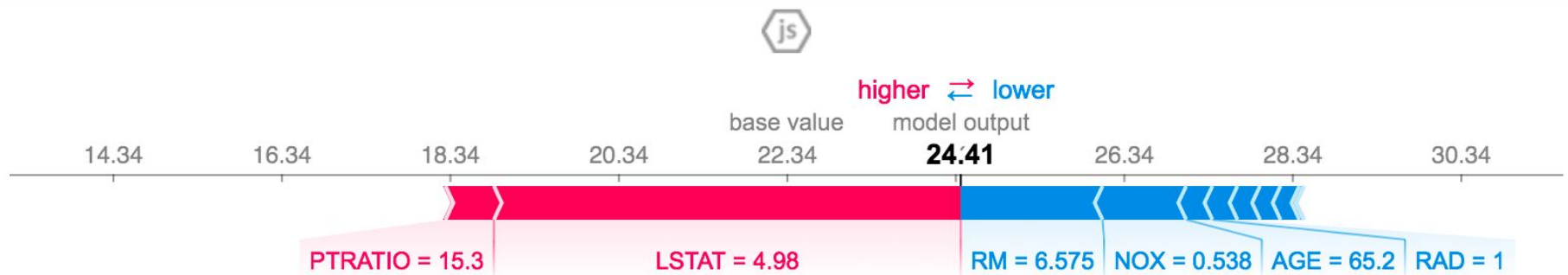
Sample data to average out features

Fix age = 40, sample other features from data

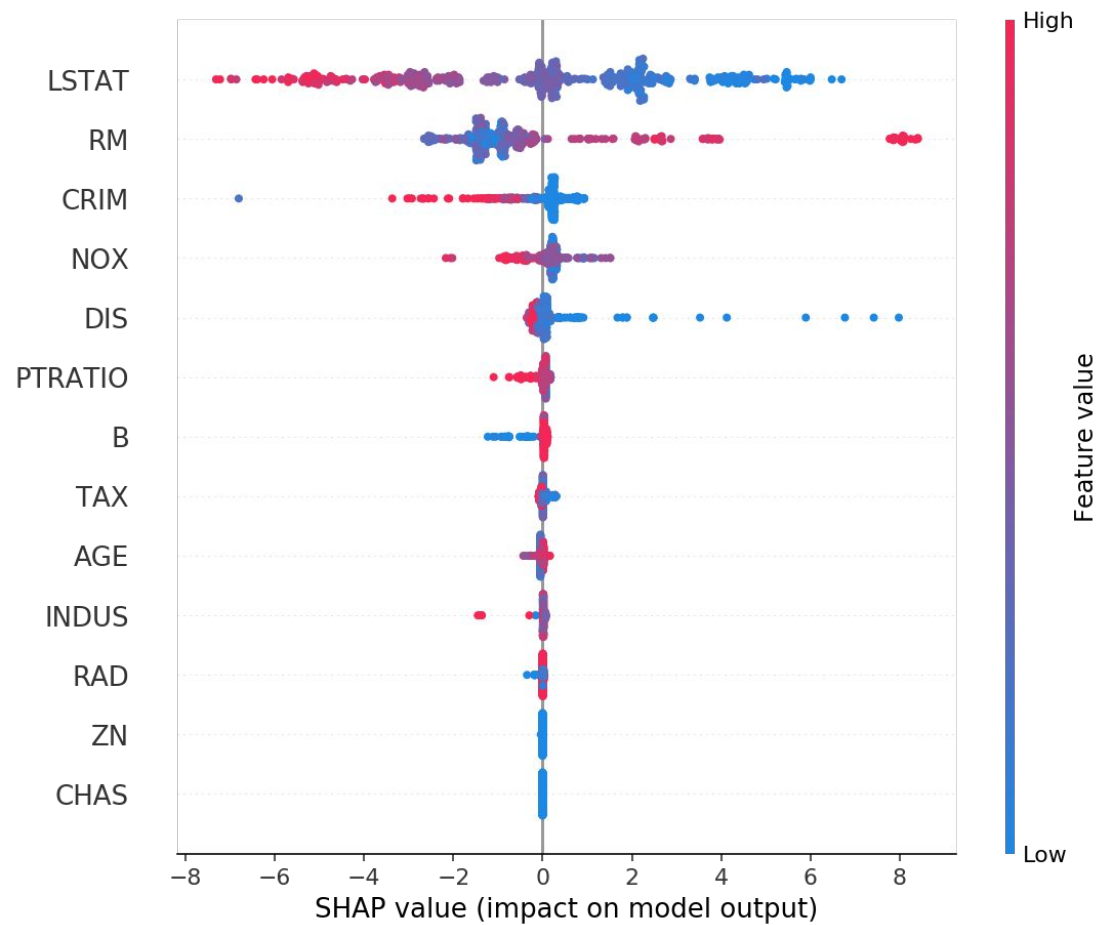
DeepExplainer, TreeExplainer



SHAP example



SHAP example



Notes on additive attribution

If the features are correlated, the contribution will be divided

F_1 = temperature

F_2 = wind speed

F_3 = wind speed/2

$$\text{PM2.5level} = 200 - 5 * F_1 - 3 * F_2 - 3 * F_3$$

Notes on attributions

It explains the model not that data.

It does not necessary tell how to improve.

If the explainer says the sales is low because of weak marketing

Does not mean increasing marketing will improve sales.

Notes on SHAP

Averaging out features by sampling can be a bad estimate

Example

Education level = Bachelor degree

Aged of 8, 10, 15, etc can be sampled from the features
(no causal relationship between features)

Beyond SHAP

New SHAP methods let you incorporate causal graphs into the model

Reduce SHAP values on cause-effect features

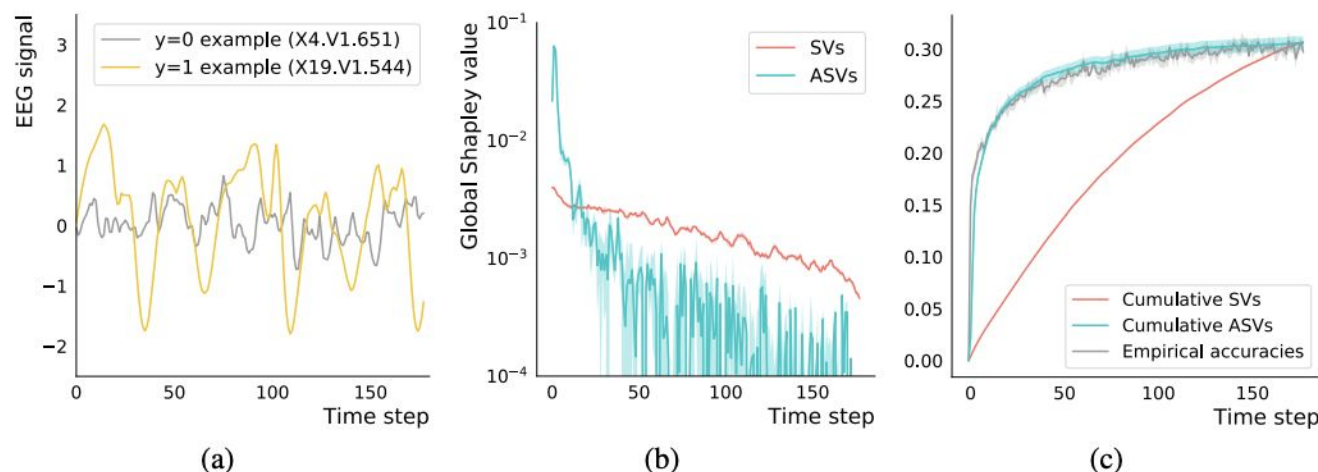


Figure 2: (a) Example EEGs: gray benign, yellow malignant. (b) Global explanations of a model trained on EEG signals. (c) Cumulative sum of the individual values in Fig. 2(b) compared to the empirical accuracy of a model trained on features $t' \leq t$.

Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability
 Christopher Frye, Colin Rowat, Ilya Feige
<https://arxiv.org/abs/1910.06358>

Lab 1

Electricity usage prediction using deep learning

Part1: Predict next day usage from day of week, previous day usage, etc.

1.1 Predict usage

1.2 Predict difference in usage

Part2: Use SHAP to study the effect of features

Confidence

ML products

Customer facing

Recommendation systems,
maps (traffic estimate),
speech2text

Best guess by the model
Mostly automatic (check deposit
by app)



Internal facing

Loan applications, demand
forecasting

Can say “I’m not sure”

Human in the loop

Machine-assisted

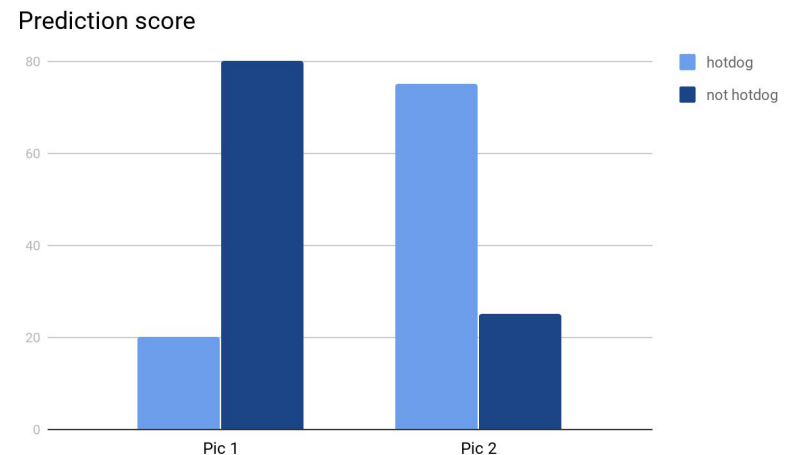
Need to say why or give
confidence level



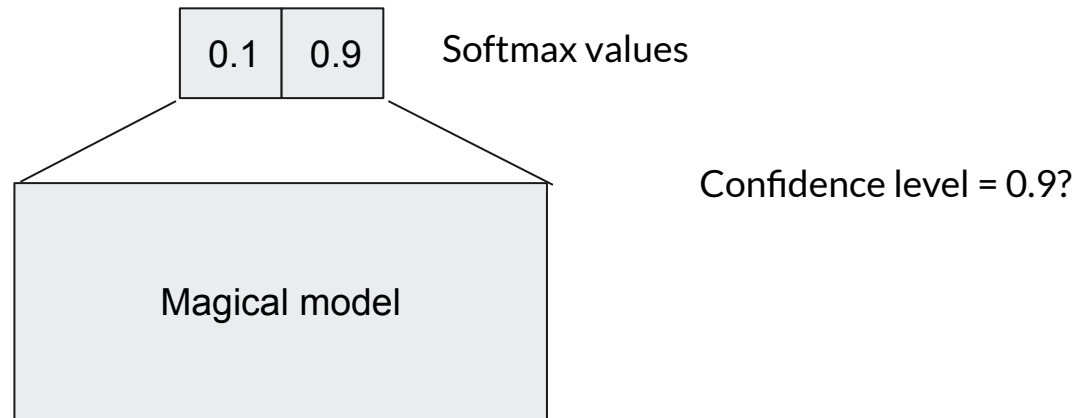
Confidence score

Practical models are not only accurate, but need to be able to state its confidence

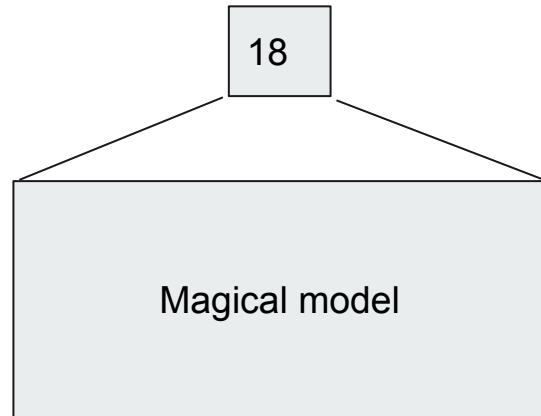
Confidence = probability of being correct



Naive way for confidence



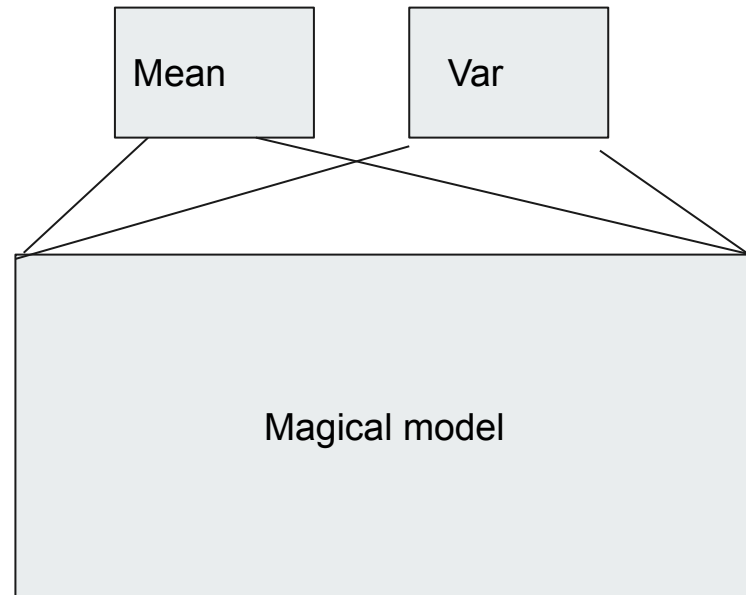
What about regression?



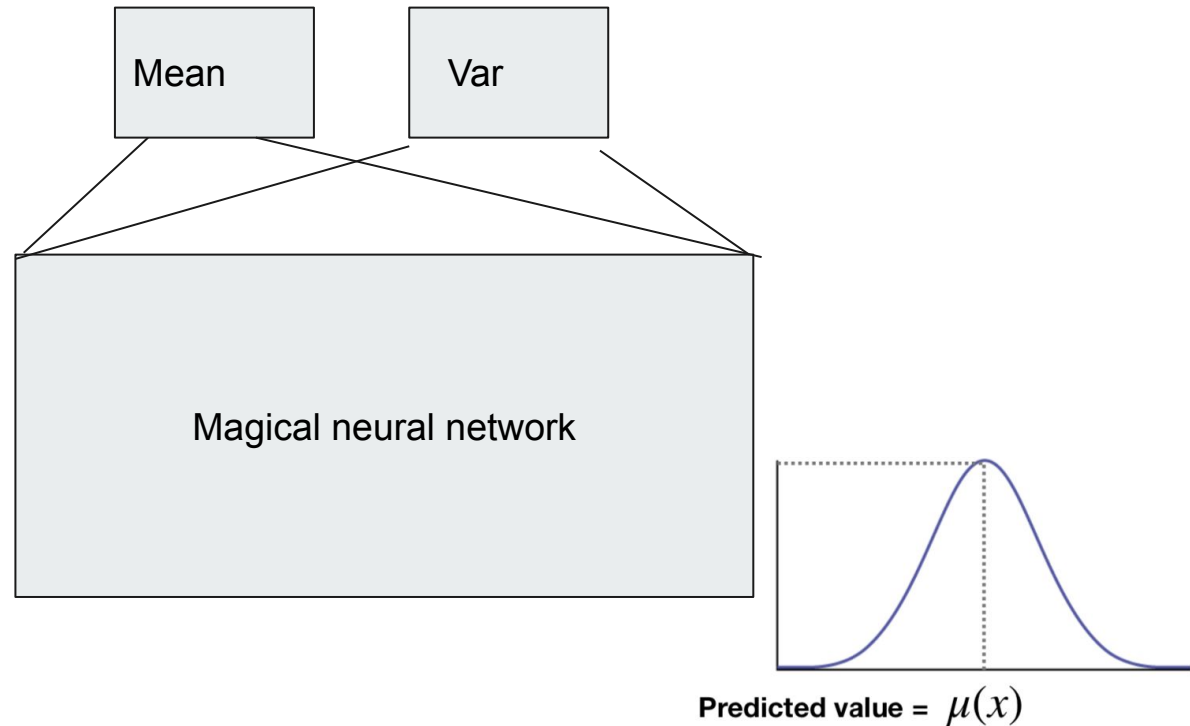
Confidence level = ???



A Naive way for regression (1994!)



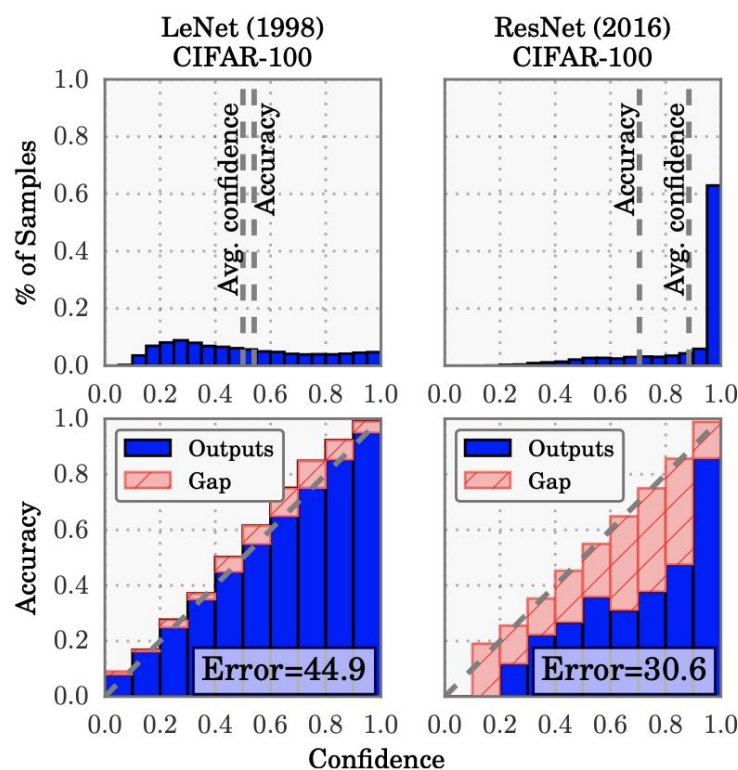
A Naive way for regression (1994!)



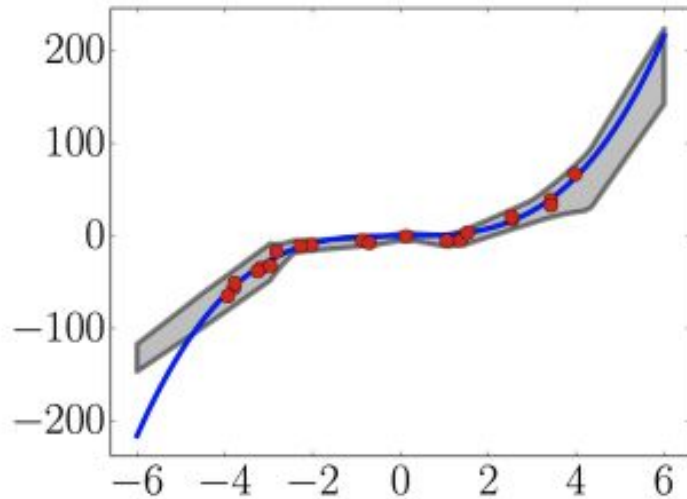
Poorly calibrated confidence

Deep Neural Networks are always overconfident!

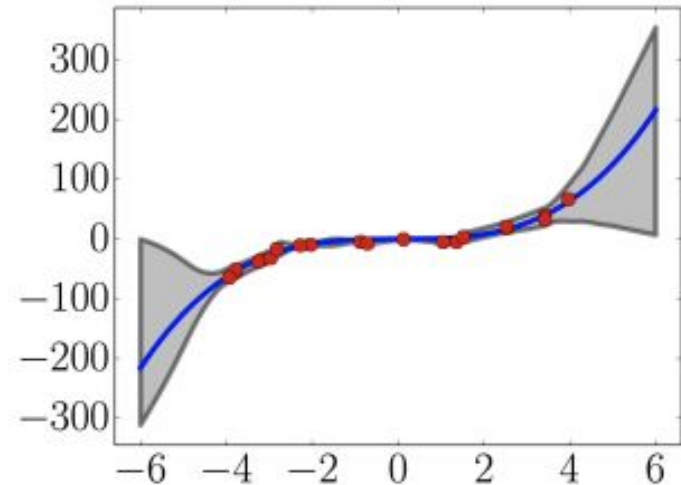
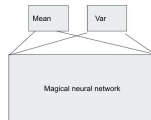
Confidence = Probability of being correct



Out of distribution problem



output from predicting variance via maximum likelihood



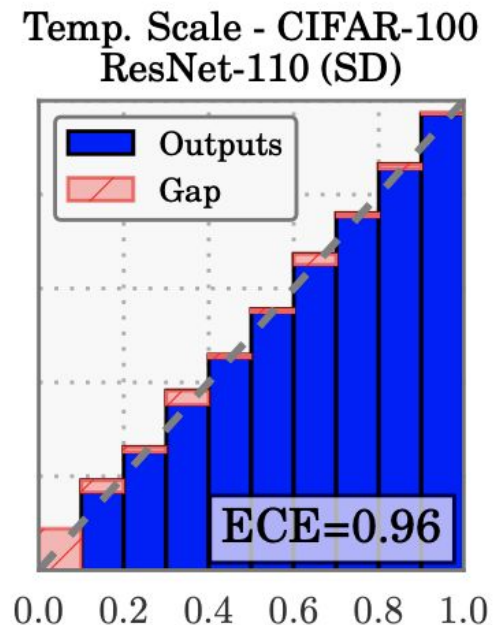
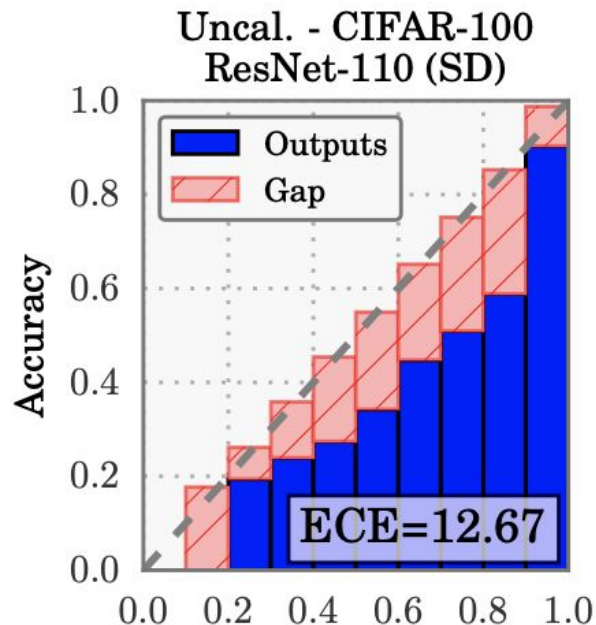
output from ensemble

Neural network calibration

Make the confidence output follows the probability of being correct.

How?

Need a separate training set to train the calibration (calibration set)



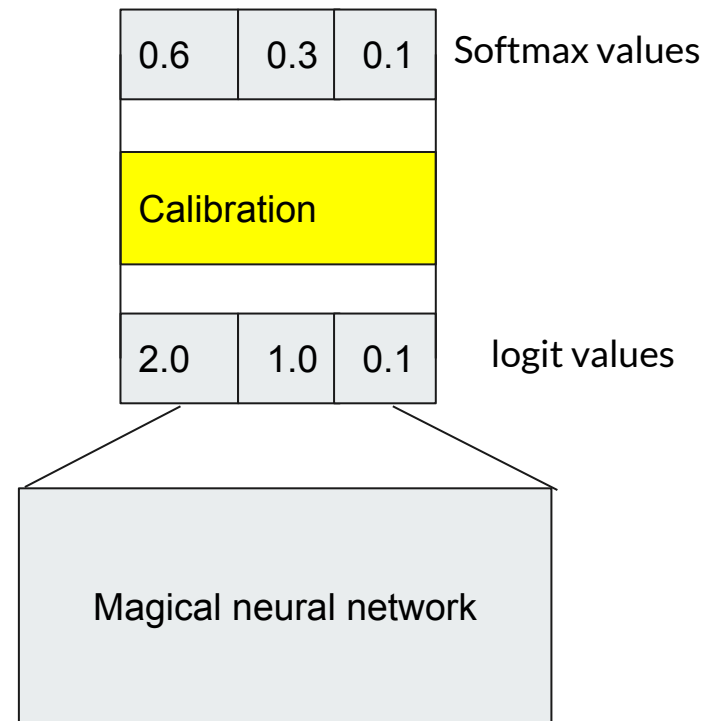
Overview of methods for calibration

1 Calibration

2 Ensemble

Calibration

Post processing after getting pre-softmax output (logit)



Calibration - Temperature scaling

Add a temperature T to rescale the softmax

$$\hat{q}_i = \max_k \sigma_{\text{SM}}(\mathbf{z}_i / T)$$

T is tuned to maximize log likelihood on the calibration set

Low T

High T



Other calibration methods

Histogram binning

Bayesian binning into quartiles (BBQ)

Matrix and vector scaling (model on top of model)

Isotonic regression (model on top of model)

Try different methods on your dataset. No absolute best.

Model ensemble for variance estimation

Intuition: if you have two independent experts, if they agree you can be confident about their judgement

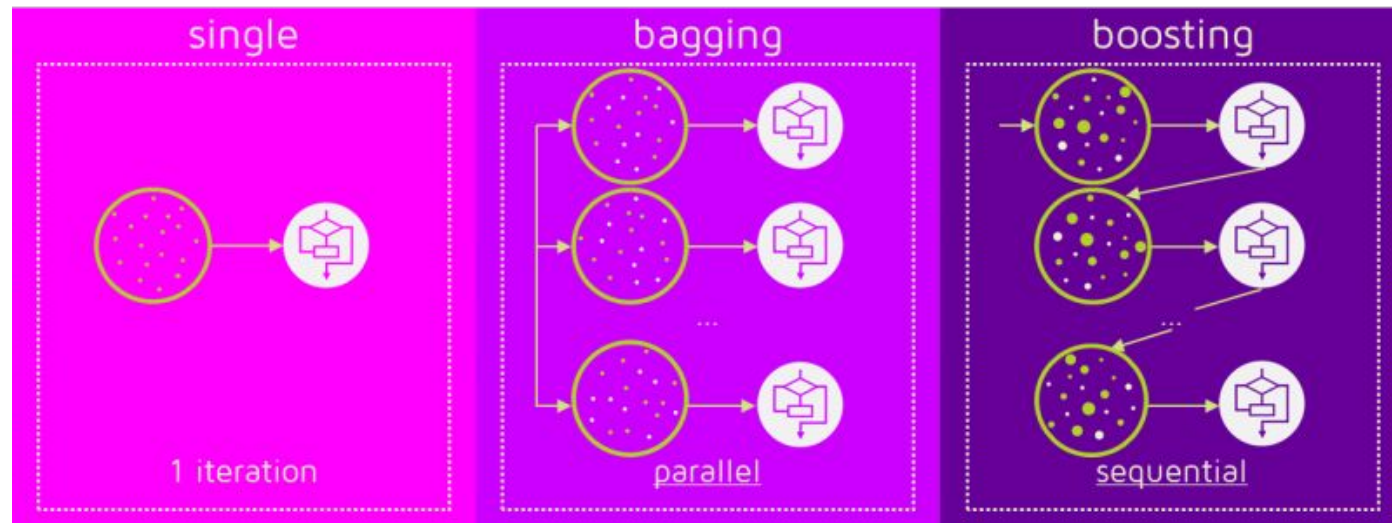


Model ensemble for variance estimation

1. Train multiple models and have them predict the same input.
2. Calculate the mean and variance of the prediction
3. The confidence can be from calculating the area under the Gaussian pdf

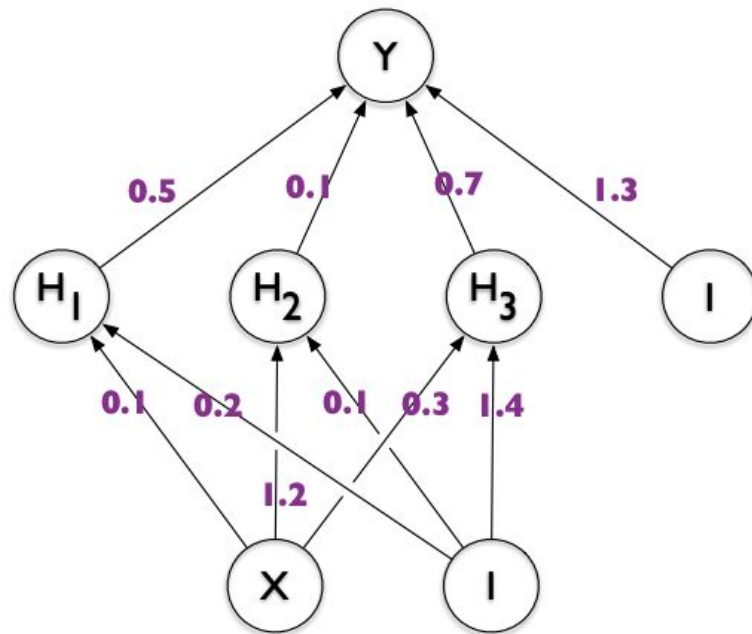
Training multiple models?

Random forest are inherently random. Training the model multiple times will give different models.

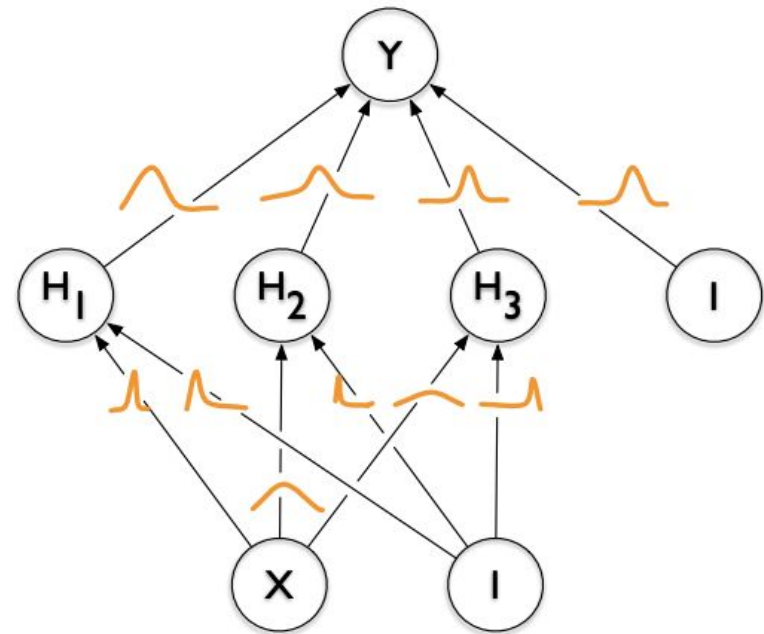


Bayesian Neural Network

Normal NN



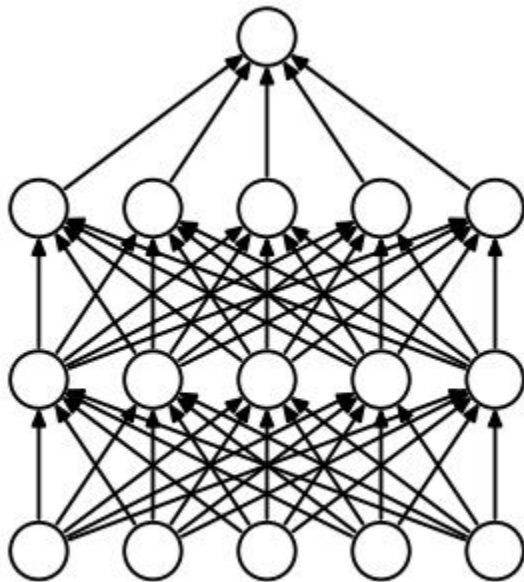
Bayesian NN



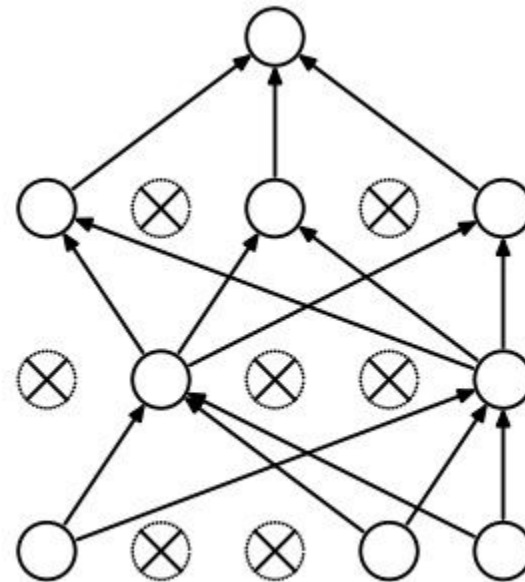
Problem: hard to do inference. Need to find $P(y | x, D) = \sum_{\theta} P(y | x, \theta) P(\theta | D)$

Training multiple models

Dropout can be used for creating multiple prediction from one neural network model



(a) Standard Neural Net

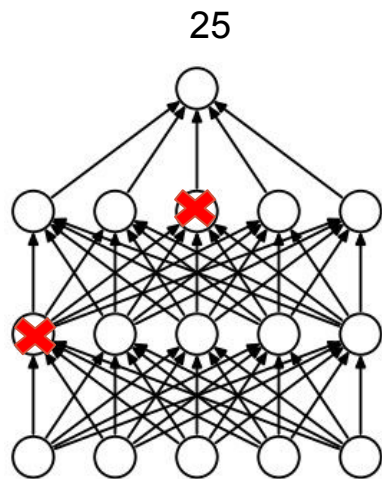


(b) After applying dropout.

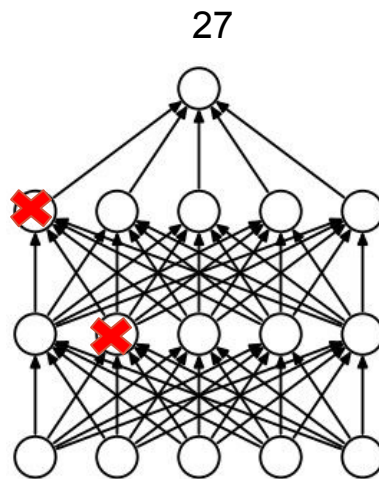
Monte Carlo dropout for confidence estimation

compute mean
and variance for
the answer and
confidence

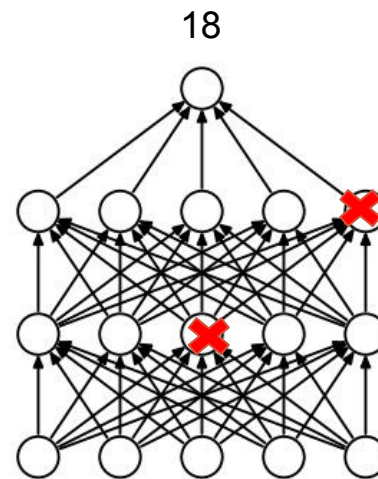
20.2 \pm 3.2



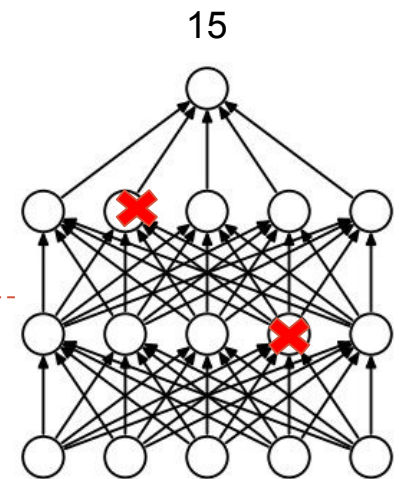
(a) Standard Neural Net



(a) Standard Neural Net



(a) Standard Neural Net



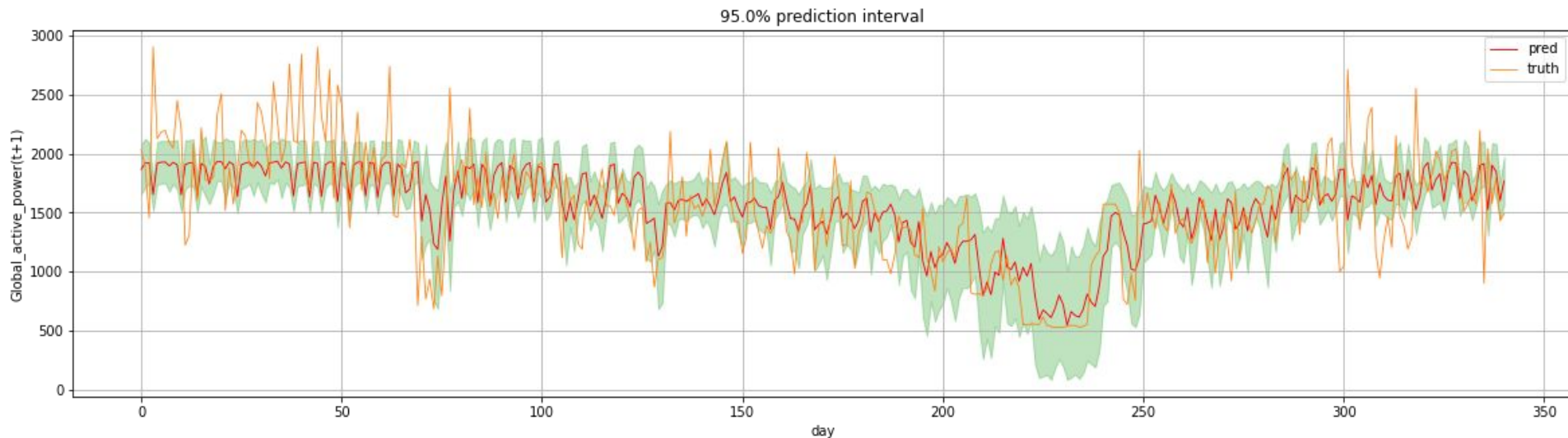
(a) Standard Neural Net

Lab 2

Let's create confidence intervals for the electricity usage

Part 1: XGBoost model

Part 2: Deep learning model



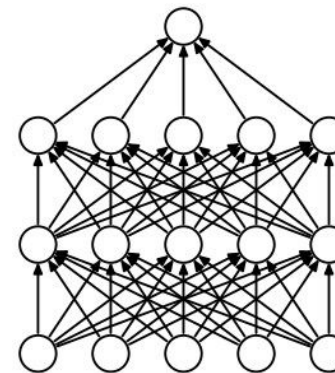
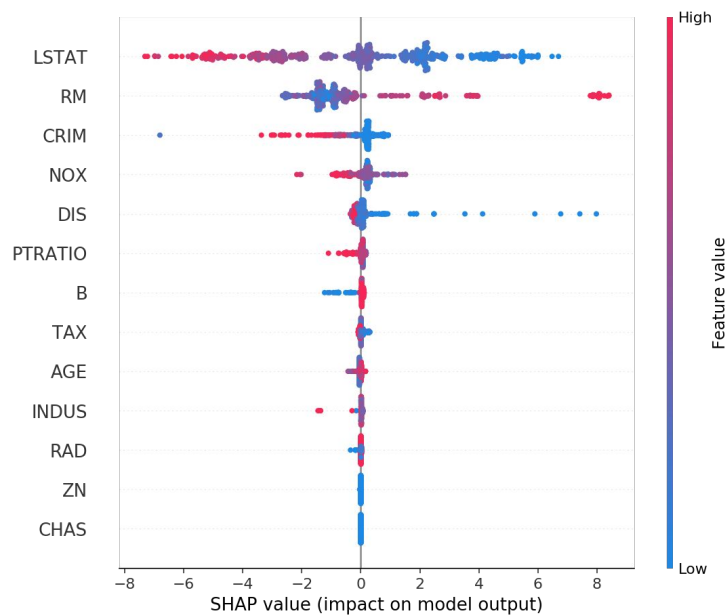
NGBoost

Use Natural Gradient instead of regular SGD (you can think of it as a kind of regularization)

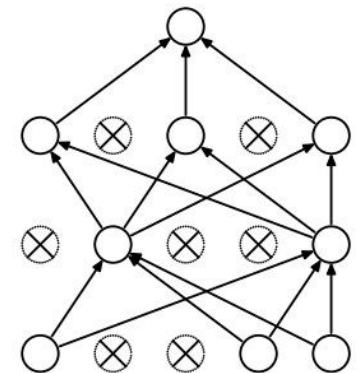
Predict mean and variance of a Gaussian and maximize the likelihood

Dataset	N	RMSE			NLL		
		MC dropout	Deep Ensembles	NGBoost	MC dropout	Deep Ensembles	NGBoost
Boston	506	2.97 ± 0.85	3.28 ± 1.00	2.94 ± 0.53	2.46 ± 0.25	2.41 ± 0.25	2.43 ± 0.15
Concrete	1030	5.23 ± 0.53	6.03 ± 0.58	5.06 ± 0.61	3.04 ± 0.09	3.06 ± 0.18	3.04 ± 0.17
Energy	768	1.66 ± 0.19	2.09 ± 0.29	0.46 ± 0.06	1.99 ± 0.09	1.38 ± 0.22	0.60 ± 0.45
Kin8nm	8192	0.10 ± 0.00	0.09 ± 0.00	0.16 ± 0.00	-0.95 ± 0.03	-1.20 ± 0.02	-0.49 ± 0.02
Naval	11934	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-3.80 ± 0.05	-5.63 ± 0.05	-5.34 ± 0.04
Power	9568	4.02 ± 0.18	4.11 ± 0.17	3.79 ± 0.18	2.80 ± 0.05	2.79 ± 0.04	2.79 ± 0.11
Protein	45730	4.36 ± 0.04	4.71 ± 0.06	4.33 ± 0.03	2.89 ± 0.01	2.83 ± 0.02	2.81 ± 0.03
Wine	1588	0.62 ± 0.04	0.64 ± 0.04	0.63 ± 0.04	0.93 ± 0.06	0.94 ± 0.12	0.91 ± 0.06
Yacht	308	1.11 ± 0.38	1.58 ± 0.48	0.50 ± 0.20	1.55 ± 0.12	1.18 ± 0.21	0.20 ± 0.26
Year MSD	515345	8.85 ± NA	8.89 ± NA	8.94 ± NA	3.59 ± NA	3.35 ± NA	3.43 ± NA

Conclusion



(a) Standard Neural Net



(b) After applying dropout.