Interpretation of Neural Networks

Enguerrand Horel

Institute for Computational and Mathematical Engineering - Stanford University

02/06/2017

Why do we care about model interpretability?

Objectives of interpretability as defined in[Lip16][1]:

- **Trust and transferability**: confidence that a model will perform well and generalize well to new data.
- **Causality and informativeness**: infer properties or generate hypotheses about the natural world.

[1] Zachary C Lipton et al. "The Mythos of Model Interpretability". In: *IEEE Spectrum* (2016).

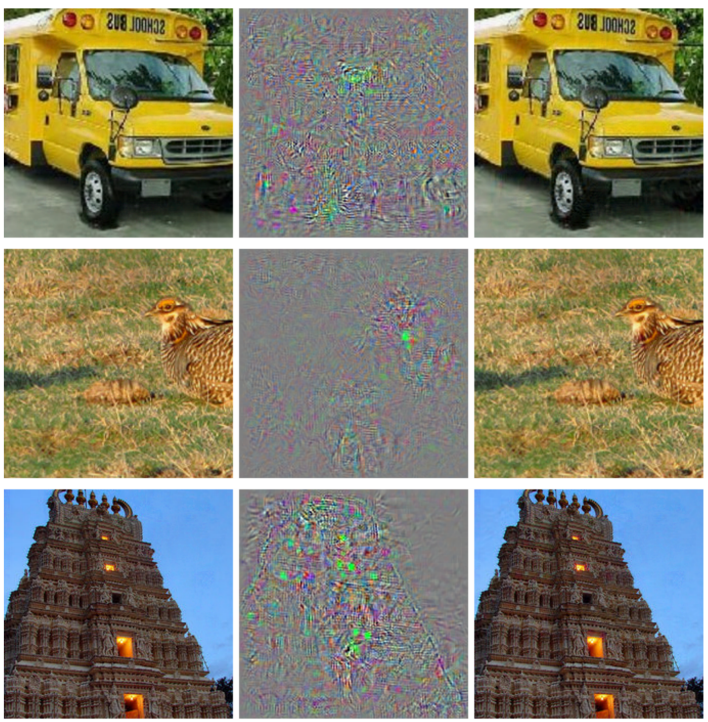
Example 1

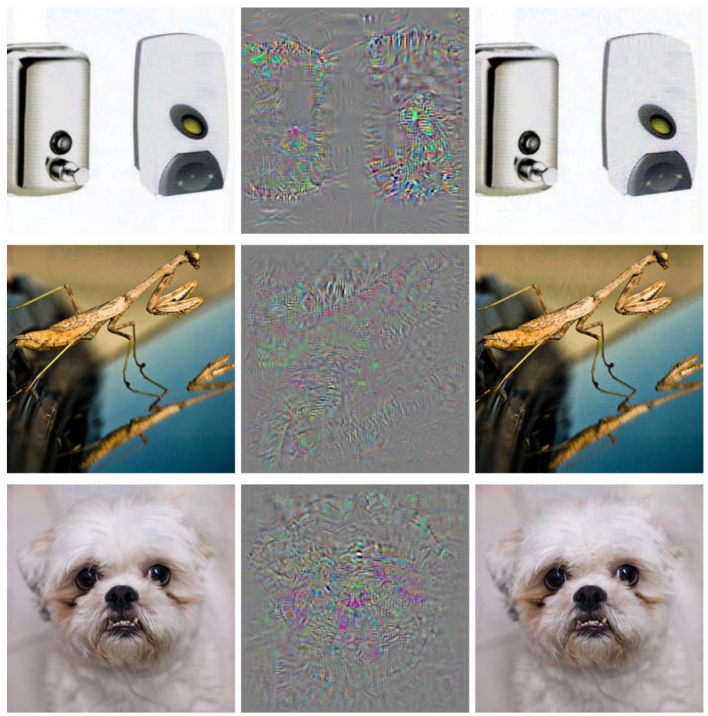
- In [Car15][2]: use of neural networks to **predict the probability of death for patients with pneumonia** so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients.
- Neural networks **outperformed traditional methods** such as logistic regression by wide margin.
- They realized that the model had learned that **patients with pneumonia who have a history of asthma have lower risk of dying from pneumonia** than the general population.
- Patients with a history of asthma who presented with pneumonia usually were admitted not only to the hospital but directly to the intensive care unit where the care they received was so effective that it lowered their risk of dying from pneumonia compared to the general population.
- However, **asthmatics have much higher risk** if not hospitalized.

[2] Rich Caruana et al. "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 1721–1730.

Example 2

- In [Sze13][3], they **cause the network to misclassify an image by applying a certain hardly perceptible perturbation**, which is found by maximizing the network's prediction error.
- The same perturbation can cause a different network, that was trained on a different subset of the dataset, to misclassify the same input.



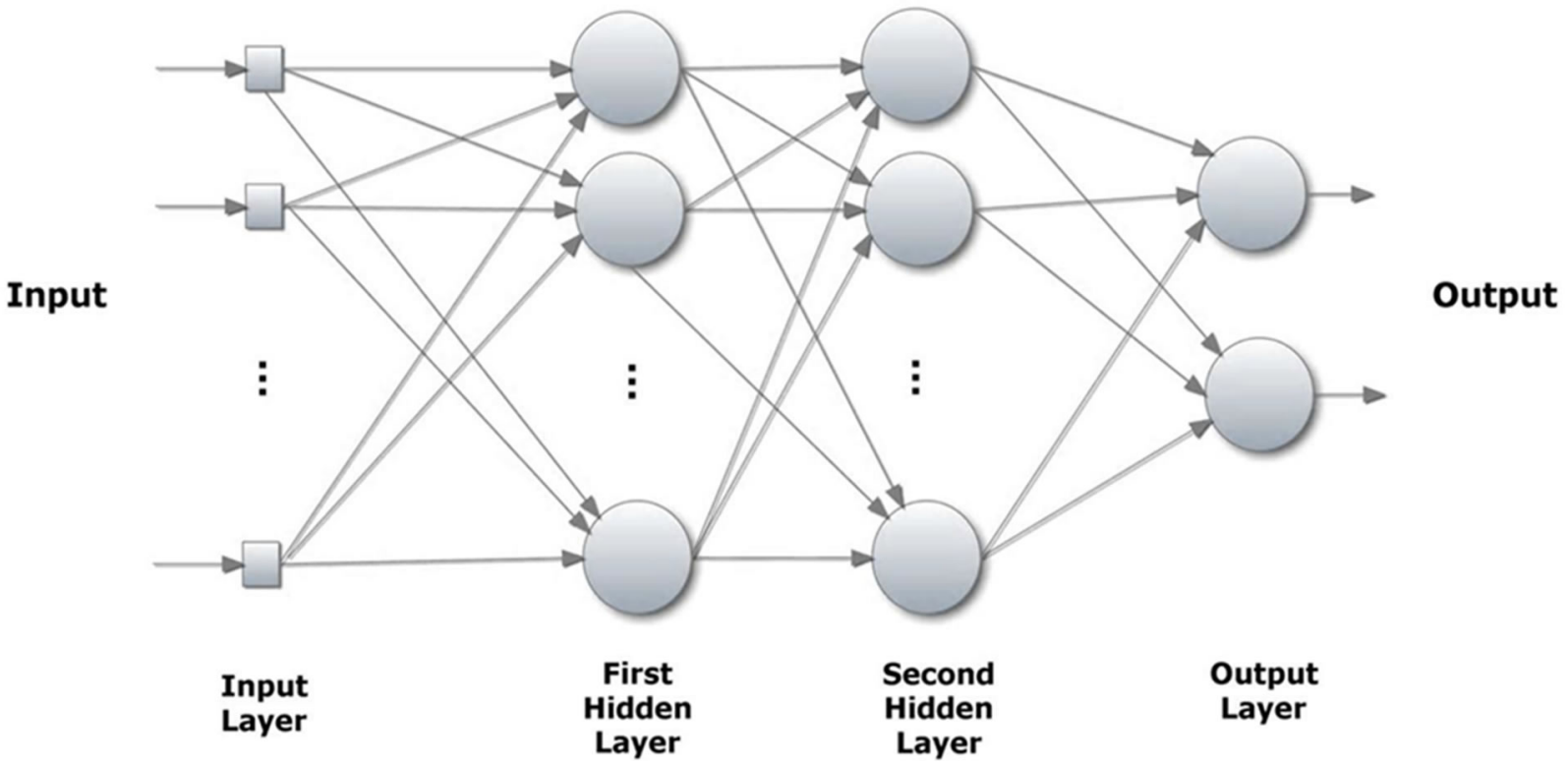


[3] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

Properties of interpretable models

- **Transparency**
  - **Simulatable:** if a person can contemplate the entire model at once, take the input data together with the parameters of the model and in reasonable time step through every calculation required to produce a prediction.
  - **Decomposability:** each part of the model - each input, parameter, and calculation - admits an intuitive explanation.
  - **Algorithmic Transparency:** what are we converging to during training?
- **Post-hoc Interpretability**
  - **Text Explanations:** e.g. this tumor is classified as malignant because to the model it looks a lot like these other tumors
  - **Visualization:** e.g. plotting gradients of predictions with respect to inputs
  - **Explanation by example:** report in addition to predictions which other examples the model considers to be most similar.

Multilayer Feedforward Neural Networks



$$f(x) = \sigma_2(W_2\sigma_1(W_1x + w_1) + w_2)$$

$x \in \mathbb{R}^p$, $W_1 \in \mathbb{R}^{k,p}$, $w_1 \in \mathbb{R}^k$, $W_2 \in \mathbb{R}^{k,m}$, $w_2 \in \mathbb{R}^m$, $f(x) \in \mathbb{R}^m$, σ_1 and σ_2 are two non-linear functions.

$\theta = \{W_1, w_1, W_2, w_2\}$ defines the set of parameters of the neural network

Order of magnitudes in financial applications

- In most of financial applications: **tens of features**.
- In [Gie16][4], they trained a neural network on a dataset of over 120 million US mortgages with hundreds of features.
- Leads to the choice of **5 hidden layers**, with **200 units** in the first hidden layer and **140 units** in each subsequent one, this represents 100K parameters.

[4] Kay Giesecke, J Sirignano, and A Sadhwani. *Deep learning for mortgage risk*. Tech. rep. Working paper, Stanford University, 2016.

Neural networks as universal approximators

- **Model-free** approximation:

$$Y = F(X) + \epsilon$$

  $F(X)$: unknown true function, X: input matrix, Y: dependent variable, ϵ: a zero mean random noise.
- Neural network estimation: $f(X; \hat{\theta})$ with $\hat{\theta} = arg\,\min_\theta L(Y, f(X; \theta))$
- In [Hor89][5], it is shown that: multilayer feedforward networks with as few as one hidden layer using arbitrary activation functions are **capable of approximating any Borel measurable function** from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available.
- However, the mapping function represented by a network is not perfect due to **suboptimal network architecture**, the **local minima** problem and the finite sample data in neural network training.

[5] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

Specification of model architecture

- **Model architecture**: number of hidden layers, number of units per layer and activation functions, subset of weights that are non-zero.
- Ideally, network architecture contains as few hidden units and connection as necessary to satisfy a good trade-off between estimation bias and variability.
- Usual approaches:
  - **grid search and cross-validation**: exhaustive search over a subset of the hyperparameter space guided by generalization performance estimated by cross-validation;
  - **regularization**: the network weights are chosen such that they minimize the error function plus a penalty term for the network complexity;
  - **pruning**: remove the weights that does not significantly increase the network error when set to zero;
  - **stopped training**: dataset split into training and validation set, training algorithm stopped when the model error in the validation set starts to grow.

Toward a more statistical approach

- Problems of these methods: **heuristic derivation**, contain a strong judgmental component
- Taking a statistical perspective is especially important for **atheoretical models** because the reason for applying them is the lack of knowledge about an adequate functional form.
- When based on on a clearly defined decision rule, model selection becomes more comprehensible and reconstructible.

Statistical model selection

- In [And99][6], the authors develop a model selection strategy based on statistical concepts: **hypothesis testing** and information criteria.
- To perform tests, the **asymptotic distribution** of the network parameters is necessary. In the one hidden layer case, and if the optimum parameter $\theta^\star$ is **at least locally unique**, it can be shown[7]

$$\sqrt{N}(\hat{\theta} - \theta^\star) \sim N(0, C)$$

with the covariance matrix C derived using the theory of misspecified model.

[6] Ulrich Anders and Olaf Korn. "Model selection in neural networks". In: *Neural Networks* 12.2 (1999), pp. 309–323.

[7] Halbert White. "Some asymptotic results for learning in single hidden-layer feedforward network models". In: *Journal of the American Statistical Association* 84.408 (1989), pp. 1003–1013.

Statistical model selection

- **Over-parametrized networks** will likely contain **irrelevant hidden units**, the weights pointing to those units can then take any particular value and are therefore not unique.
- The loss function will present **flat regions** and gradient based algorithms trying to minimize it may output irrelevant estimates.
- **Bottom-up approach**: to avoid over-parametrization and obtain statistically valid results, need to begin with an empty model and successively add hidden units.

Statistical model selection

- Previous results are true only in the **one hidden layer** case and are based on **asymptotic distributions**.
- **Estimation of the actual distribution** of the parameters in the general case could be done using **bootstrap**.
- Bootstrap suffers from the presence of **multiple** local and global optima, i.e. lack of model **identifiability**.

Neural network identifiability

- **Identifiability**: different values of the parameters must generate different models.
- However with neural networks, some transformations of the set of parameters can generate a new network similar to the initial in terms of overall input-output map and hence to the same predictive performance.
- In [Chen93][8], they defined **equioutput transformations**: $g : \Theta \to \Theta$ a mapping from parameter space to parameter space which leaves the output of the network unchanged, i.e. $f(x;\theta) = f(x;g(\theta))$.
- They then show that all equioutput transformations are **compositions of interchange and sign flip transformations**.
- In the best case, neural networks are identifiable up to these equioutput transformations and have several global optima.

[8] An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. "On the geometry of feedforward neural network error surfaces". In: *Neural computation* 5.6 (1993), pp. 910–927.

Statistical model selection

- Because of the presence of these multiple minima, bootstrap tends to **overestimate the variance** of the parameters distributions and tests will fail to reject insignificance of too many parameters.
- One solution to overcome the lack of identifiability is to **restrict the parameter space** to ensure unicity of minima. But restrictions are arbitrary and could lead to wrong inference.
- Hence, we are interested in a class of estimators that are invariant to parameters transformations.
- **Parameter-transformation invariant statistic**: the same inference should be made from data and a model involving a parameter θ as would be made from the same data if the model used a parameter ϕ, where ϕ is a one-to-one transformation of θ, $\phi = h(\theta)$.

Inference after model selection

- If a researcher has mined the data (i.e. selected an empirical model based on a series of trial) inferences based on the final set of results are in general incorrect.
- For example, as explained in[Lee16][9] if one starts with a large number of candidate variables and does not know a priori which are relevant, it is tempting to let the data decide which variables to include in the model.
- One common approach is to fit a linear model with all variables included, observe which ones are significant, and then refit the linear model with only those variables included.
- The p-values can no longer be trusted, since the variables that are selected will tend to be those that are significant.
- Intuitively, we are "overfitting" to a particular realization of the data.

[9] Jason D Lee et al. "Exact post-selection inference, with application to the lasso". In: *The Annals of Statistics* 44.3 (2016), pp. 907–927.

Inference after model selection

- This problem is also called pretesting, data dredging, data fishing, data snooping.
- In [Ioa05][10], the author claims that **90% of medical research is wrong** because of invalid inference following mining.

[10] John PA Ioannidis. "Why most published research findings are false". In: *PLos med* 2.8 (2005), e124.

Overview of interpretation

- Two main approaches to interpret a trained neural network:
  - **Decompositional approach:** look into the network at the level of units and parameters
  - **Black-box approach:** consider the network purely as a mapping from input to output and try to interpret that mapping.

Rule extraction methods

- Type of rules:
  - **If-then rules**: If $X \in S(i)$ Then $Y = y(i)$
  - **M-of-N rules**: If at least M of the following N variables are true Then...
  - **Tree-structured** representation
- Large literature: [And95][11] surveys techniques for extracting rules from trained neural networks.
- Methods mostly heuristic and the trade-off between number and complexity of the rules and their accuracy and fidelity is hard to balance.
- Look for interpretations closer to traditional statistical inference.

[11] Robert Andrews, Joachim Diederich, and Alan B Tickle. "Survey and critique of techniques for extracting rules from trained artificial neural networks". In: *Knowledge-based systems* 8.6 (1995), pp. 373–389.

What would I like to know?

- Interpretable models:
  - **Linear model**: $Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p$
  - Each coefficient represents the sensitivity of the dependent variable with respect to the associated feature.
  - The linearity entails the independence of the effects of the variables.
  - **Logistic regression**:

$$log\left(\frac{P(Y=1|X)}{P(X=0|X)}\right) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p$$

  - Each coefficient represents the sensitivity of the log of the odds with respect to the associated feature.
- What is the effect of each feature on a certain function of the outcome?
- What are the interactions between the features?

Measure feature relative importance

- Number of heuristic measures have been proposed to estimate the relative importance of input features to the output variable as reviewed in [Zha00][12]:
  - **sum of the absolute input weights** (does not consider the impact of perhaps more important hidden node weights);
  - **pseudo weight**: sum of the product of weights from the input node to the hidden nodes and corresponding weights from the hidden nodes to the output node;
  - **Garson's measure**: partitions the hidden layer weights into components associated with each input node and then the percentage of all hidden nodes weights attributable to a particular input node is used to measure the importance of that input variable;
  - **rank of the variables**: based on the decrease of the accuracy of the classifier when variable is removed.
- Ideally, we would like these measures to be parameter-transformation invariant statistic.

[12] Guoqiang Peter Zhang. "Neural networks for classification: a survey". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30.4 (2000), pp. 451–462.

Gradient based methods

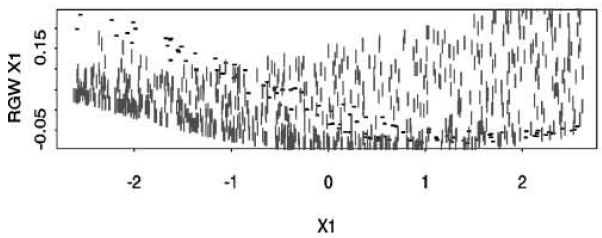
- The derivative of the prediction function with respect to one explanatory variable is a good measure of its impact on the outcome.
- Aggregated measure:

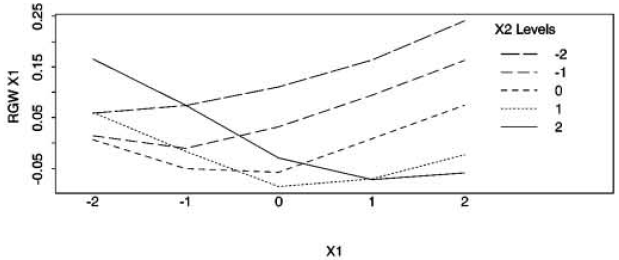
$$\tilde{\theta}_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial \hat{f}(x_i)}{\partial x_{ij}}\right)^2}$$

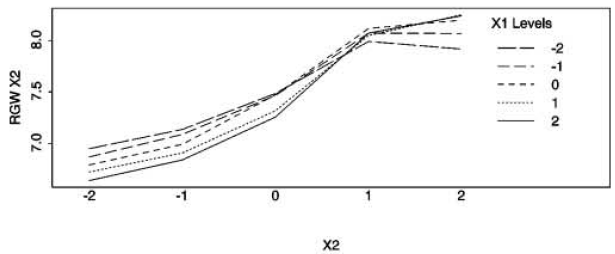
- **Scatter plots** of the gradient of each variable with respect to its values provide a mean for examining the possibility of non-linearity.
- **Split-level plots** are used to detect interactions.

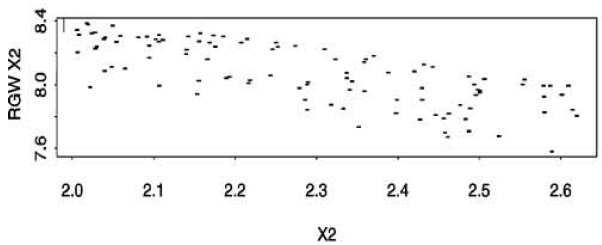
Gradient based methods

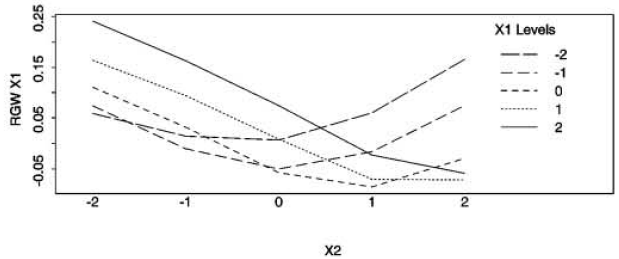
13

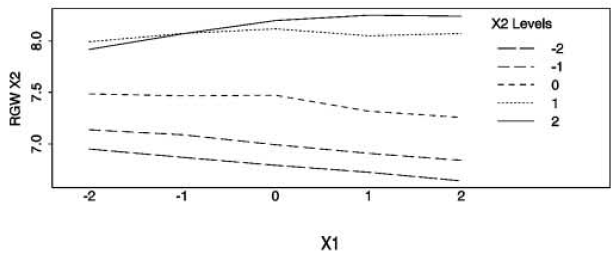












[13] Orna Intrator and Nathan Intrator. "Interpreting neural-network results: a simulation study". In: *Computational statistics & data analysis* 37.3 (2001), pp. 373–393.

Interactions between variables

- Functional ANOVA decomposition: $F(x) : \mathbb{R}^k \to \mathbb{R}$, for $u \subset \{1, ..., k\}$ we denote by x_u the subset of variables whose are indexes are in u. We can write $F(x)$ uniquely as

$$F(x) = \sum_{u \subseteq \{1,...,k\}} f_u(x)$$

  with $f_u(x)$ depending only on x_u.
- F can be represented as a constant, plus terms in one variable, plus terms in two variables and so forth.
- Let's define G_U the $\mathcal{L}_2$ projection of F onto the set of functions with ANOVA structure described by U a collection of subsets of $\{1, ..., k\}$.
- The projection of a 3 dimensional function $f(x_1, x_2, x_3)$ onto the set of functions with ANOVA structure 1,2 is given by $f_0 + f_1(x_1) + f_2(x_2)$

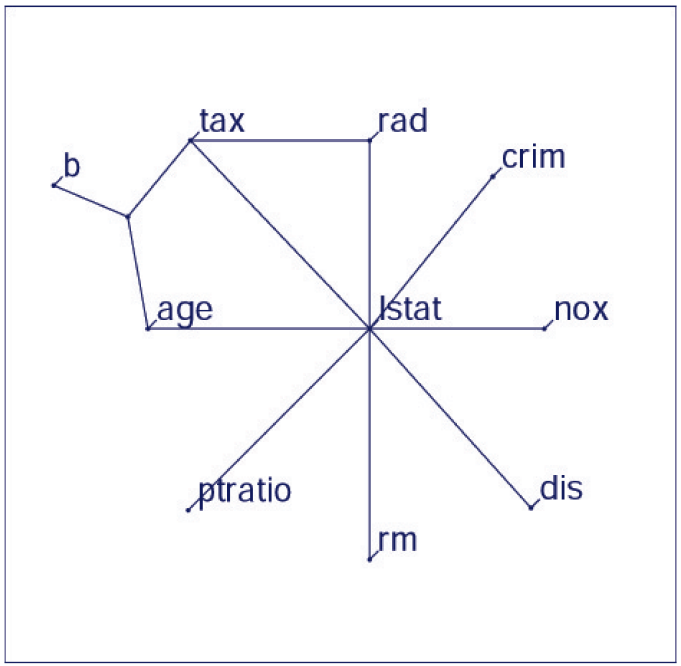
Interactions between variables

- To evaluate the significance of an interaction u, we project f onto the set of all functions with no interaction in u, G_u.
- Say that f has ANOVA structure described by a collection U of subsets of $\{1, ..., k\}$ if $f_u(x) = 0$ for all u that are proper supersets of some elements of U.
- Want to find a minimal U such that $E(F(x) - G_U(x))^2 < \epsilon$
- Efficient algorithm to find all the significant interactions described in [Hoo04][14].

[14] Giles Hooker. "Discovering additive structure in black box functions". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, pp. 575–580.

Interactions between variables

- Test on a 13-16-1 neural network trained on the Boston Housing data set.
- 13 variables: CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT, MEDV.



- Include all but three variables, all remaining variables have interactions with LSTAT apart form B. There is only one second order interaction in AGE, TAX, B.

Relearning approach

- Objective: learn in a simple way from what has been learned in a complex way by the neural network.
- The initial complex trained neural network will be called the teacher model and will teach to a simpler student model.
- In [Ba14][15], it is shown that one hidden layer neural networks (student model) are capable of learning the same function as deep nets (teacher model) when learned from the teacher model.
- Most of the time, training the small network directly on the original training set without the help of the teacher model will lead to a lower accuracy than the student model.
- Similar results of model compression are shown in [Buc06][16] and [Hin15][17].

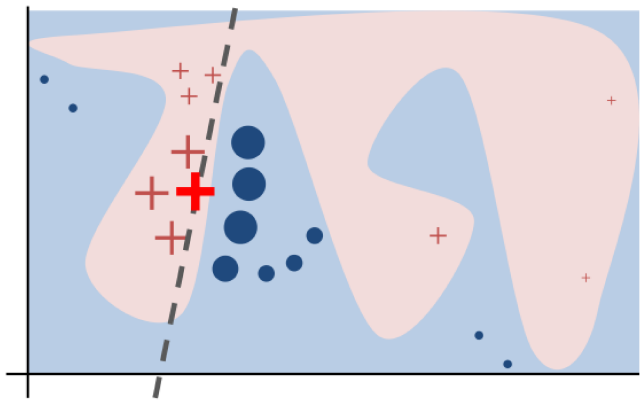
[15] Jimmy Ba and Rich Caruana. "Do deep nets really need to be deep?" In: *Advances in neural information processing systems*. 2014, pp. 2654–2662.

[16] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2006, pp. 535–541.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

Neural network approximation

- Learning a simple linear model locally around a prediction as shown in [Rib16][18].



- Global approximation using general additive model with interactions [Lou13][19]:

$$g(y) = \sum f_i(x_i) + \sum f_{ij}(x_i, x_j)$$

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *arXiv preprint arXiv:1602.04938* (2016).

[19] Yin Lou et al. "Accurate intelligible models with pairwise interactions". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 623–631.