MODEL EVALUATION IN THE LAND OF DEEP LEARNING

About me

Pramit Choudhary

@MaverickPramit

www.linkedin.com/in/pramitc/

pramit.choudhary@h2o.ai

- Lead data scientist(ML Scientist) at h2o.ai.
- Previously Lead data scientist at DataScience.com(acquired by Oracle)
- Will be continuing the work on exploring better ways to evaluate, extract and explain the learned decision policies for predictive models at h2o.ai
- For sometime I was also using machine learning algorithms to find love for eHarmony.

H2O.ai

Machine Learning workflow

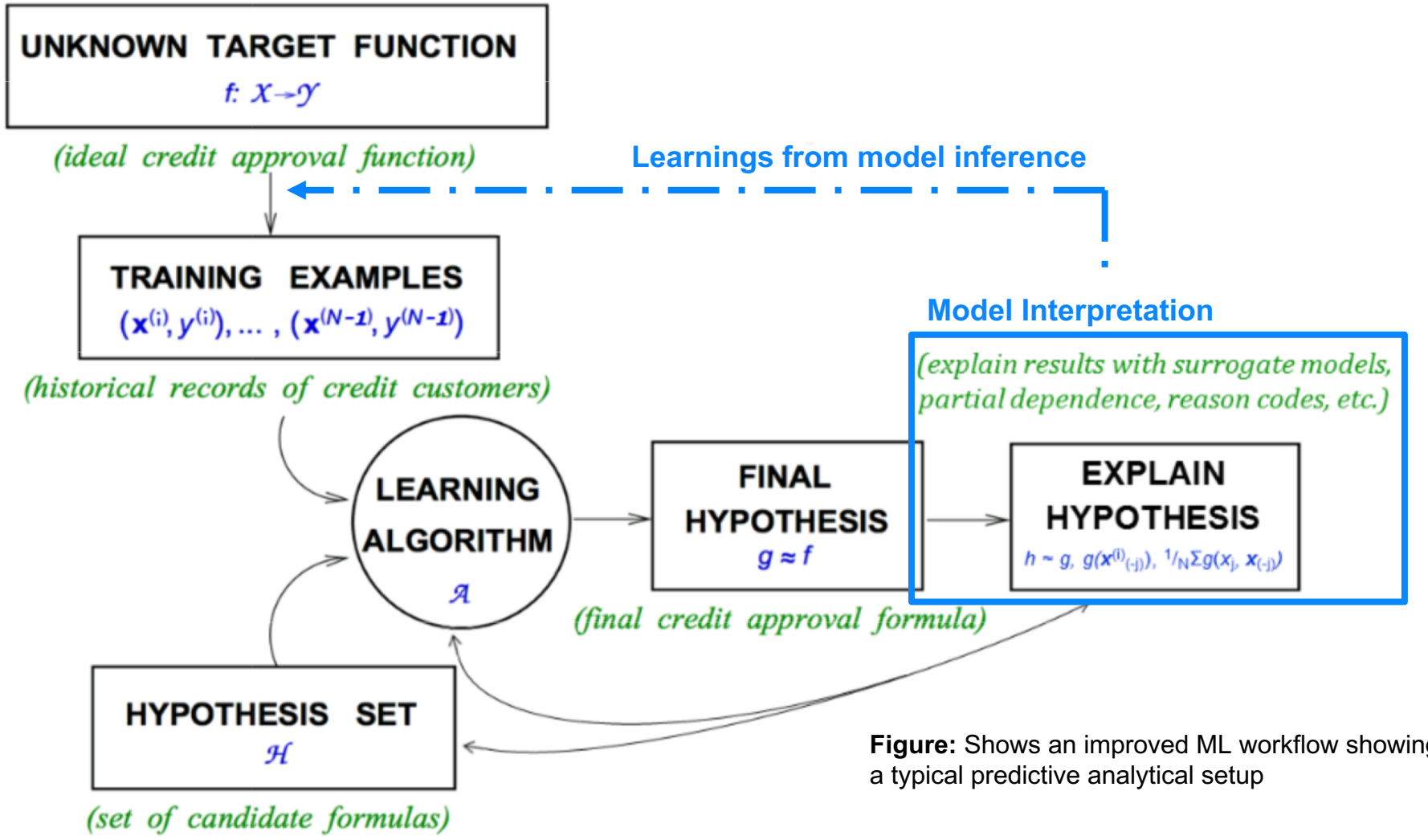


Figure: Shows an improved ML workflow showing how MLI fits into a typical predictive analytical setup

**Reference: Adapted from "Learning from data - Professor Yaser Abu-Mostafa"

H2O

What is Model Interpretation?

- Ability to explain and present a model in a way that is understandable to humans.

-- Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning.” arXiv preprint. 2017. https://arxiv.org/pdf/1702.08608.pdf

- “A collection of visual and/or interactive artifacts that provide a user with sufficient description of the model behavior to accurately perform tasks like evaluation, trusting, predicting, or improving the model.” - *Assistant Professor Sameer Singh, UCI*

- “Why did you do A” or ”Why DIDN’T you do B”, ”Why CAN’T you do C” ?

-- Gilpin, Leilani H., et al. "Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning." *arXiv preprint arXiv:1806.00069* (2018).

- A Model’s result is self descriptive and needs no further explanation; expressed in terms of inputs and outputs. “Mapping an abstract concept(e.g. predicted class) into a domain that human can make sense of“

-- Gregoire Montavon et al. “Methods for Interpreting and Understanding Deep Neural Networks”

H2O

Motives for Model Interpretation

Model Maker(Producer)

1. Helps in defining hypothesis for the problem
2. Debugging and improving an ML system – mismatched objectives.
3. Exploring and discovering latent or hidden feature interactions (useful for feature engineering/selection and resolving preconceptions).
4. Understanding model variability to avoid over-fitting.
5. Helps in model comparison.
6. Building domain knowledge about a particular use case.
7. Bring transparency to decision making to enable trust and safety – ML System is making sound decisions.

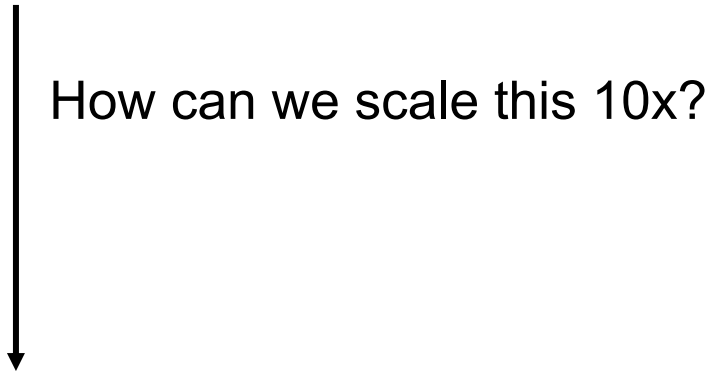
Model Breaker(Consumer)

1. Ability to share the explanations to consumers of the predictive model.
2. Explain the model/algorithm.
3. Explain the key features driving the KPI.
4. Verify and validate the accountability of ML learning systems, e.g., causes for false positives in credit scoring, insurance claim frauds(identifying spurious correlations is easier for humans).
5. Identify blind spots to prevent adversarial attacks or fix dataset errors.
6. Comply with data protection law and regulations, e.g., EU's GDPR.

How will it help?

Currently

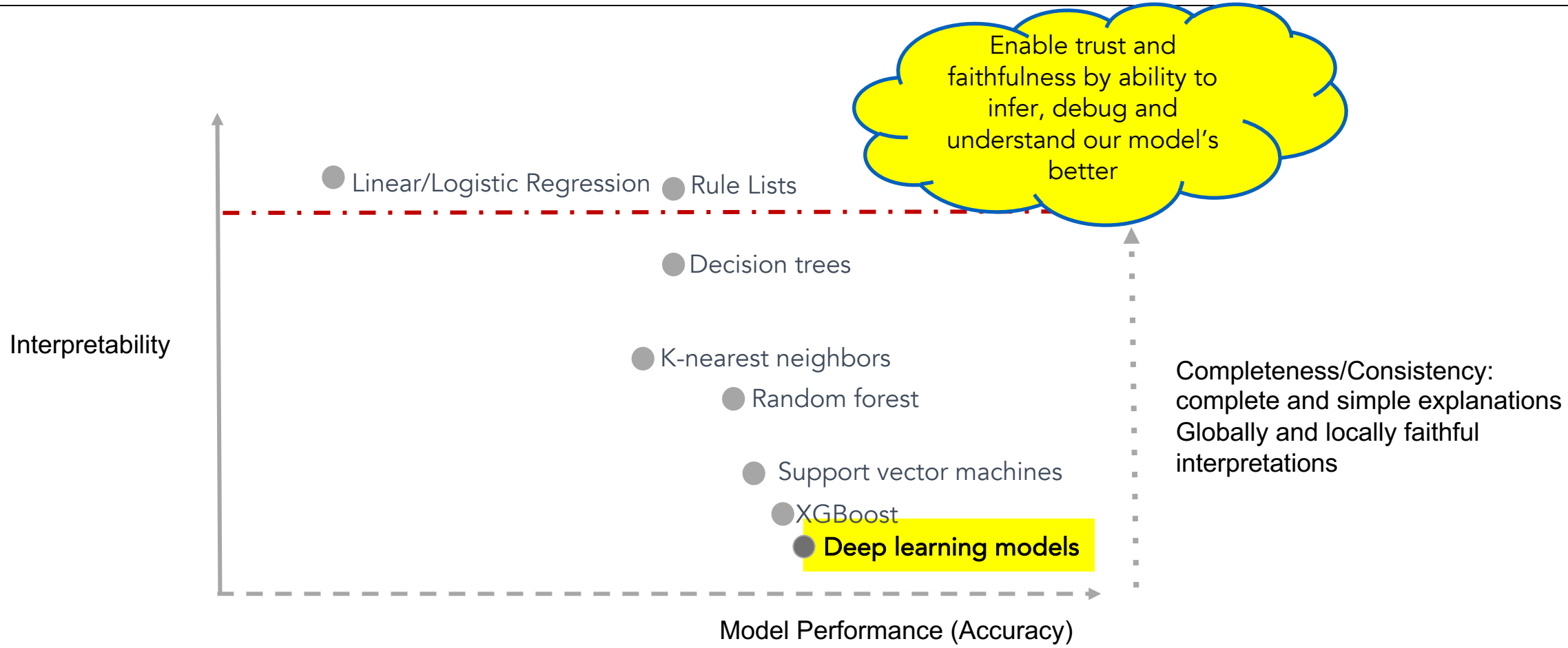
ML use-case = data munging + data scientist expertise + model building with availability to computation



Driverless Machine Learning + Machine Learning Interpretation(MLI)

ML use-case = data munging + data scientist expertise + model building with availability to computation

Uncover the impressive Facade



"While model interpretation is a hard problem, it's within the role of the data scientist to guide the other stakeholders through different levels of interpretation, recognize the caveats, highlight ambiguities, etc" - Paco Nathan(derwen.ai)

H2O

Scope of Interpretation

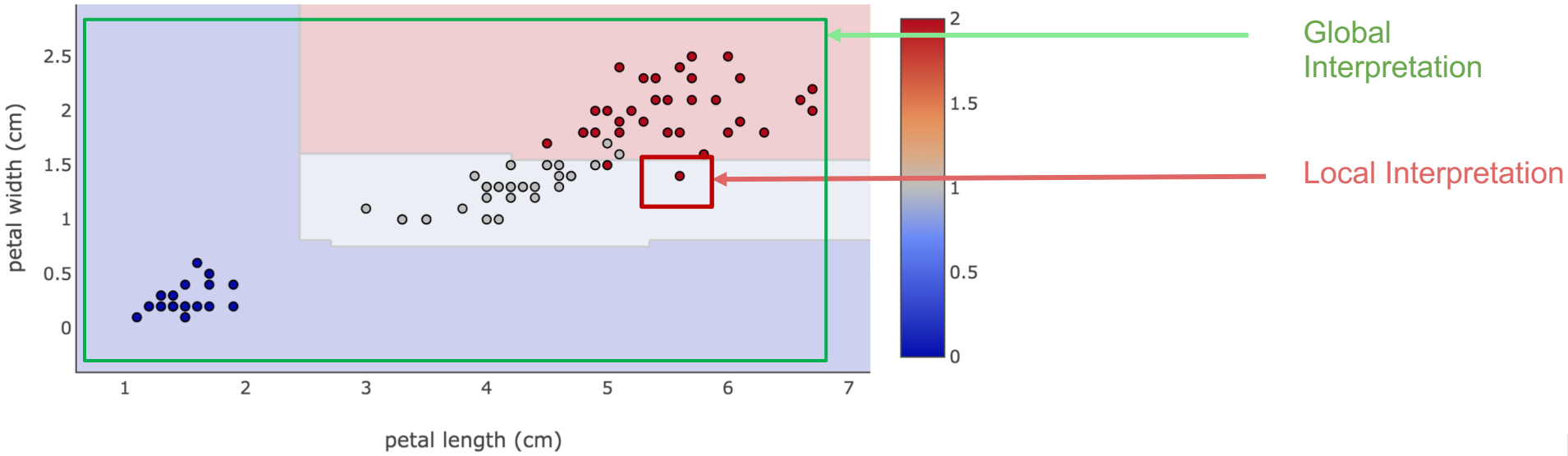
Global Interpretation

Being able to explain the conditional interaction between dependent (response) variables and independent (predictor or explanatory) variables based on the complete dataset. Helpful in explaining the context of the decision classification.

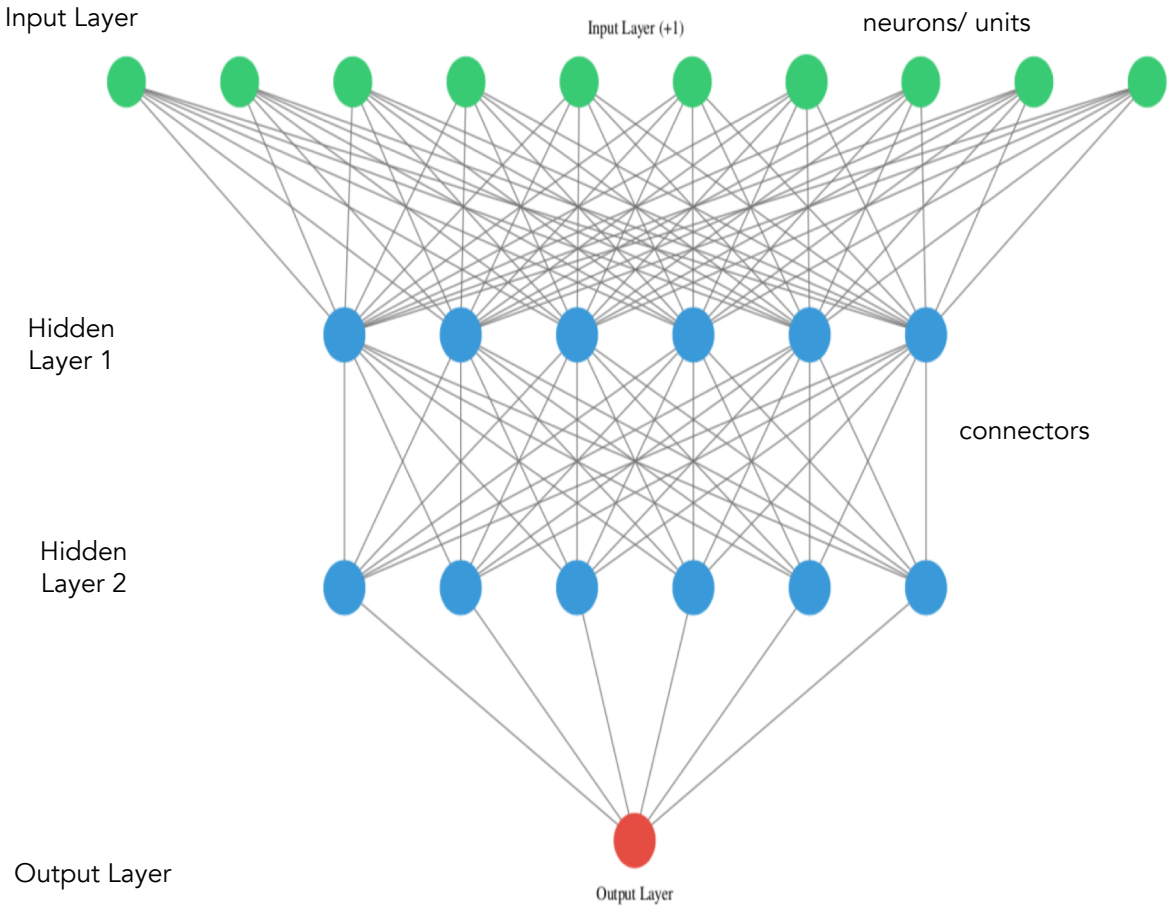
Local Interpretation

Being able to explain the conditional interaction between dependent (response) variables and independent (predictor or explanatory) variables for a single row or a subset of rows. Helpful in identifying local trends and intuitions.

$$y = f(x, w)$$



Deep Neural Network



- Helps build expressive and flexible models by learning arbitrary non-linear and non-convex functions easily.
- Can be expressed in different architectures; optimizing for accuracy and computation efficiency often leads to complex designs.
- Need for manual feature engineering is less; lower layers can extract complex features; and better regularization strategies.
- With advancement in software - Keras/Tensorflow/MXNet and hardware - better integration with GPU, it's easier to train DNN's over billions of data points optimizing over large number of parameters.
- But, models are often perceived as "black boxes" because of lack of tools to infer them.

H2O

Modern DNNs with complex designs

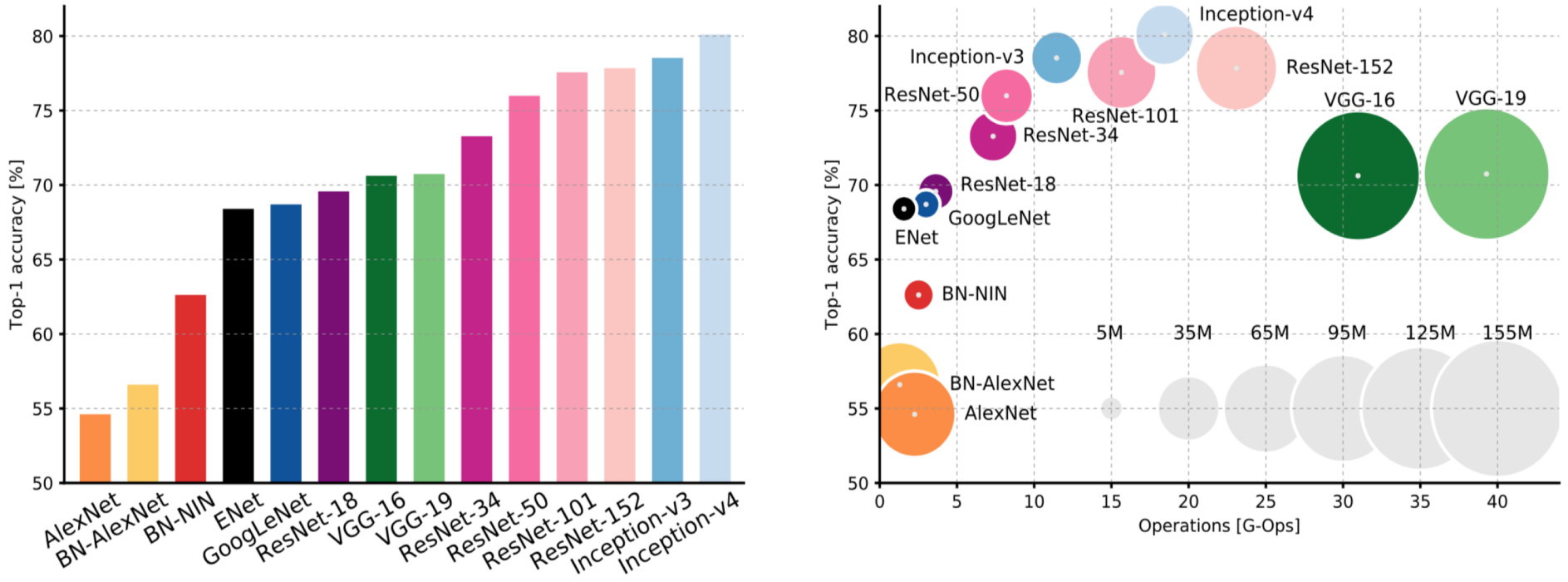


Image Source: Canziani, Alfredo, Paszke, Adam & Culurciello, Eugenio (2016). An Analysis of Deep Neural Network Models for Practical Applications

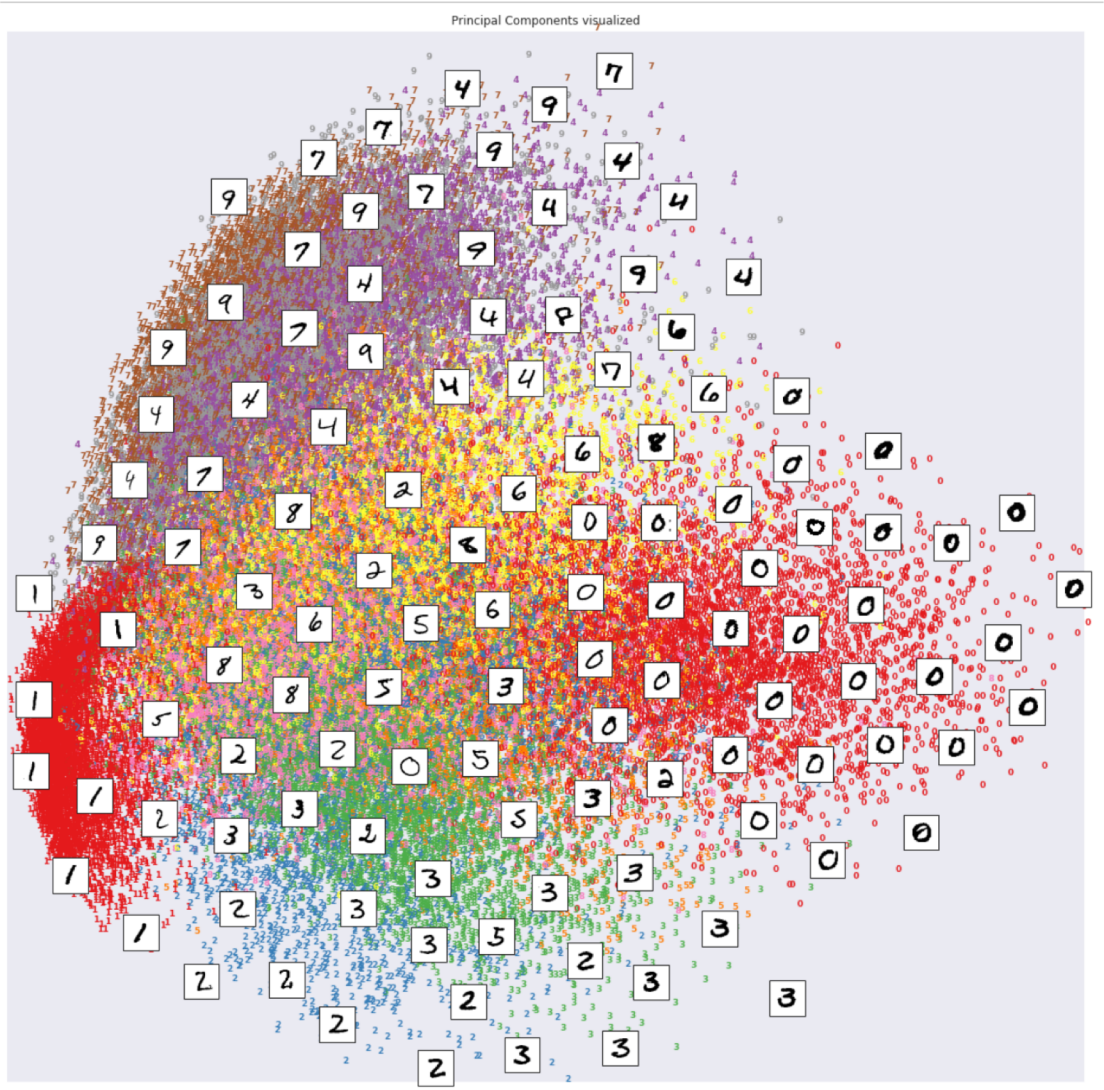
Optimizing on accuracy and similar related metrics, has made Modern DNN architectures complex and often difficult to interpret and understand as humans.

H2O

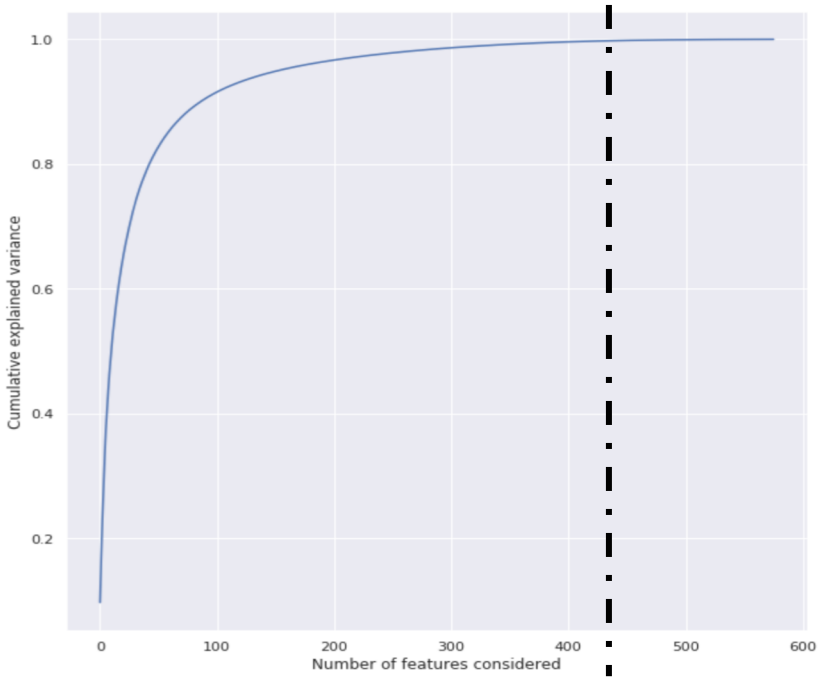
Interpretability Algorithms

Identifying Novel Representation: Dimensionality Reduction

- Mapping data points from high-dimensional space to low dimensional space is very helpful in identifying patterns in the input data.



A

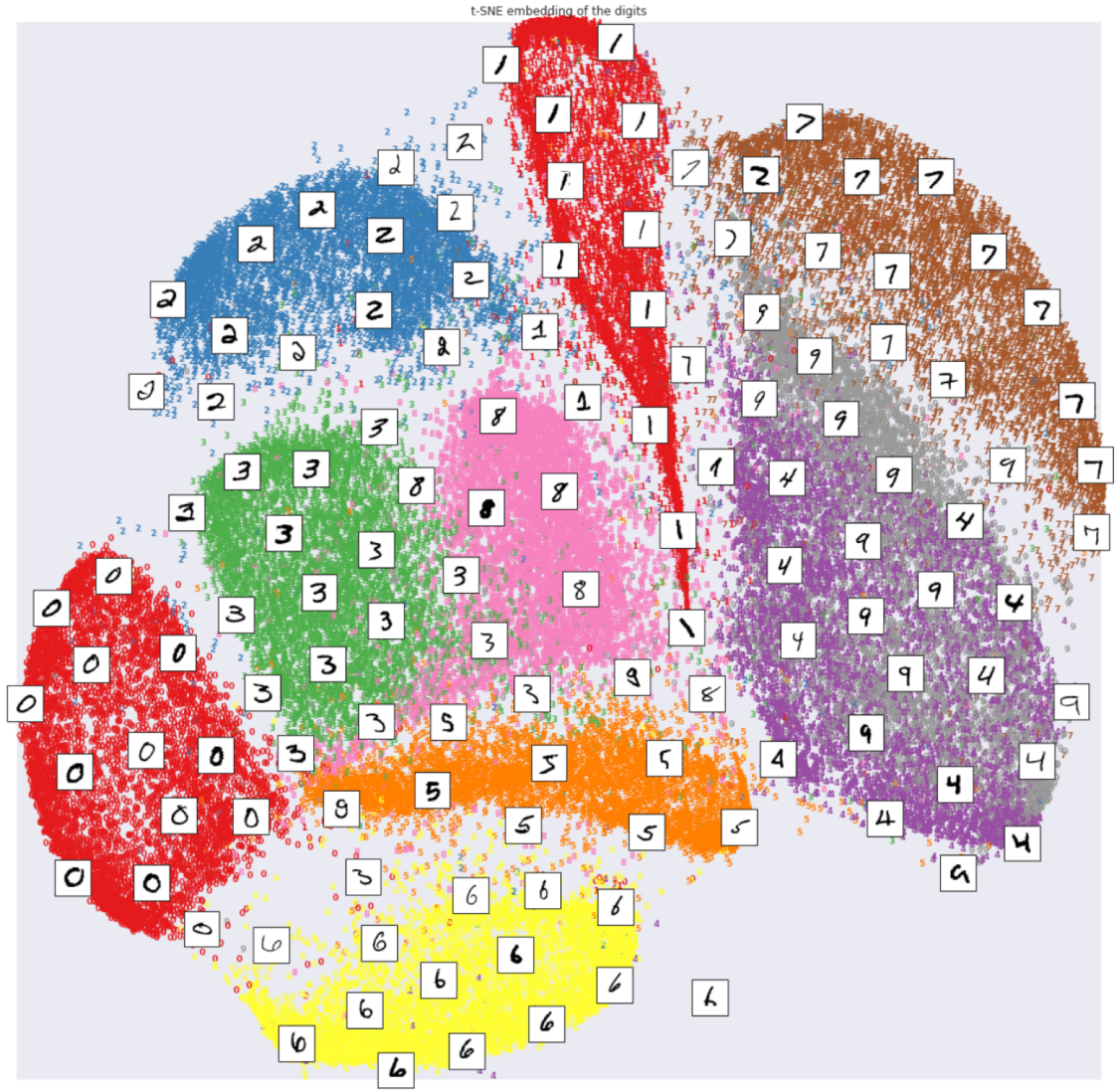


B

- **Figure A:** 2D PCA projection is applied on the MNIST datasets for representing high-dimensional space in a lower dimension in such a way that variance is maximized.
- Is a linear dimensionality reduction technique.
- Dissimilar data points in high dimensional space may get in-correctly represented in low dimensional space.
- Original image is represented using 784 dimensions vs the reduced dimension of ~450 capturing 100% variation.
- **Figure B:** plots the cumulative variance vs the number of principle components.

H_2O

Manifold Learning

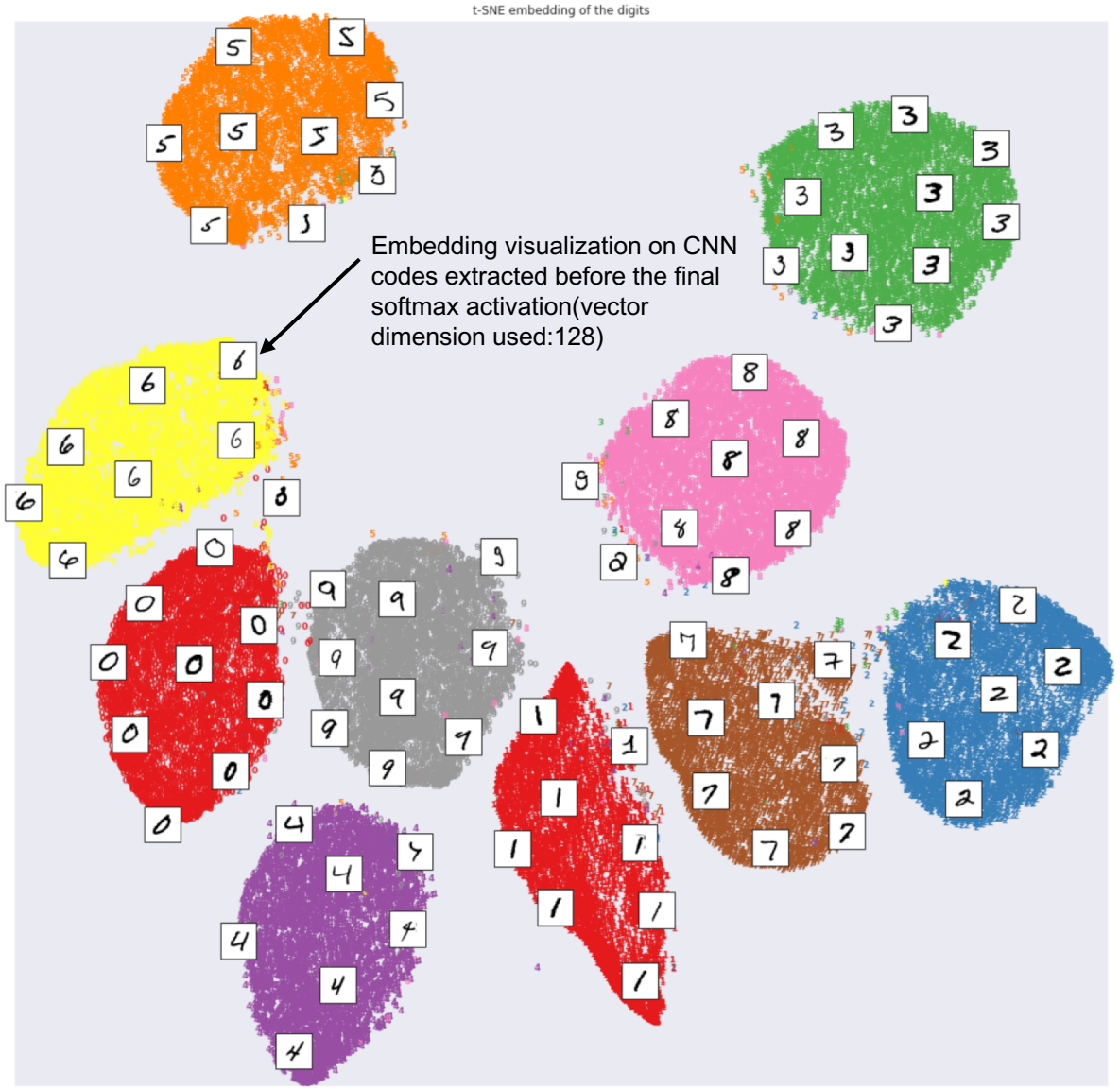


- t-distributed stochastic neighbor embedding (t-SNE) by van der Mateen and Hinton(pronounced "tee-snee") (http://lvdmaaten.github.io/publications/papers/MachLearn_2012.pdf)
- Is a popular non-linear dimensionality reduction technique
- Maps N high dimensional data vectors ($X = \{x_1, \ldots, x_n\}$) to low-dimensional embedding $Y = \{y_1, y_2, \ldots, y_n\}$
- Converts pairwise similarities between high-dimensional data points (x_i and x_j) into joint probability distribution(P) over all pairs of non-identical points.
- Cost function: perplexity(*consider a value between 5 and 50*). Choice of this parameter is not extremely critical.
- Optimizing parameters: no. of iterations(*should be at-least 250*); learning rate(*range [10.0, 1000.0]*)
- "How to Use t-SNE Effectively" http://distill.pub/2016/misread-tsne/
- Other dimensionality techniques: Self-organizing map(SOM), Isomap, LLE(Local Linear Embedding), autoencoders

Feature codes with t-SNE

A

Embedding visualization on raw features before model fitting



B

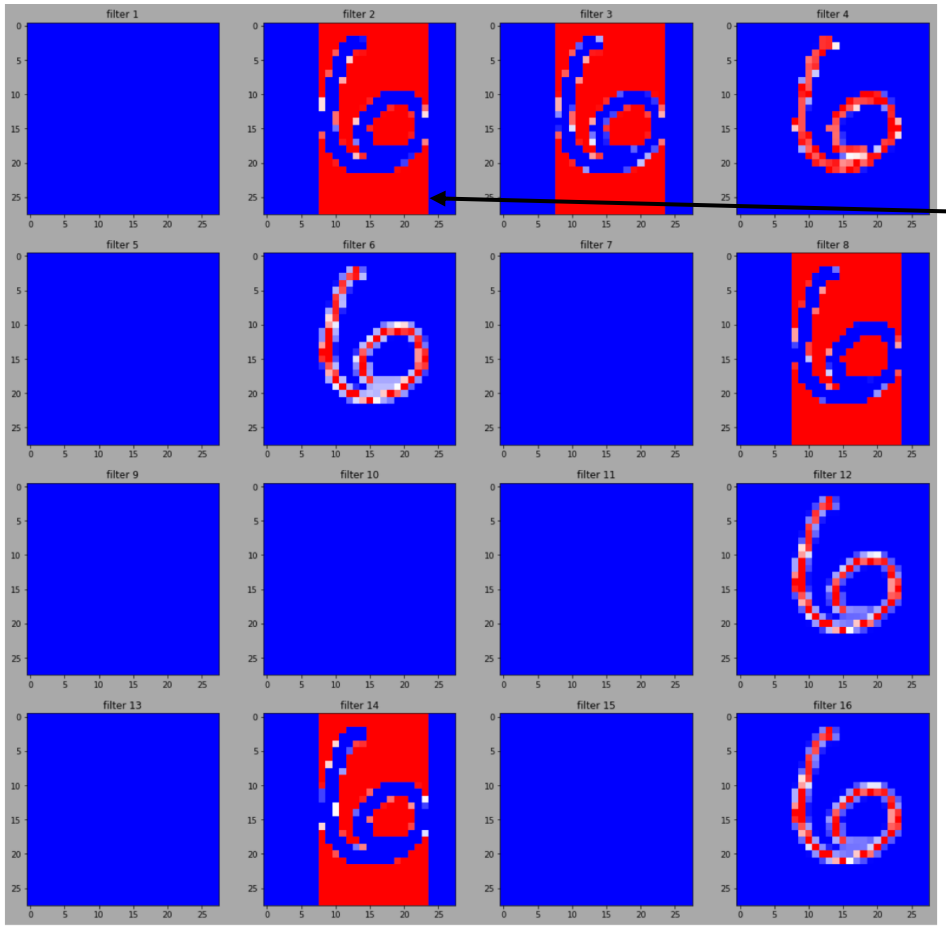
Embedding visualization on CNN codes extracted before the final softmax activation(vector dimension used:128)

- CovNets could approximately be interpreted by embedding the images into a low dimensional representation where the class categories are linearly separable
- Since, linear decomposition to lower dimension does not optimize on actual separation in high-dimensional space. t-SNE could be a good choice here.
- Tries to minimize the KL divergence between the join probability of the low dimensional embedding and high-dimensional data points
- **Figure A** represent t-SNE computed on the raw MNIST data points.
- **Figure B** represent t-SNE embedding using the extracted feature codes from the CNN model. Images that appear close to each other are most likely close in the CNN's learned representation as well.

H2O

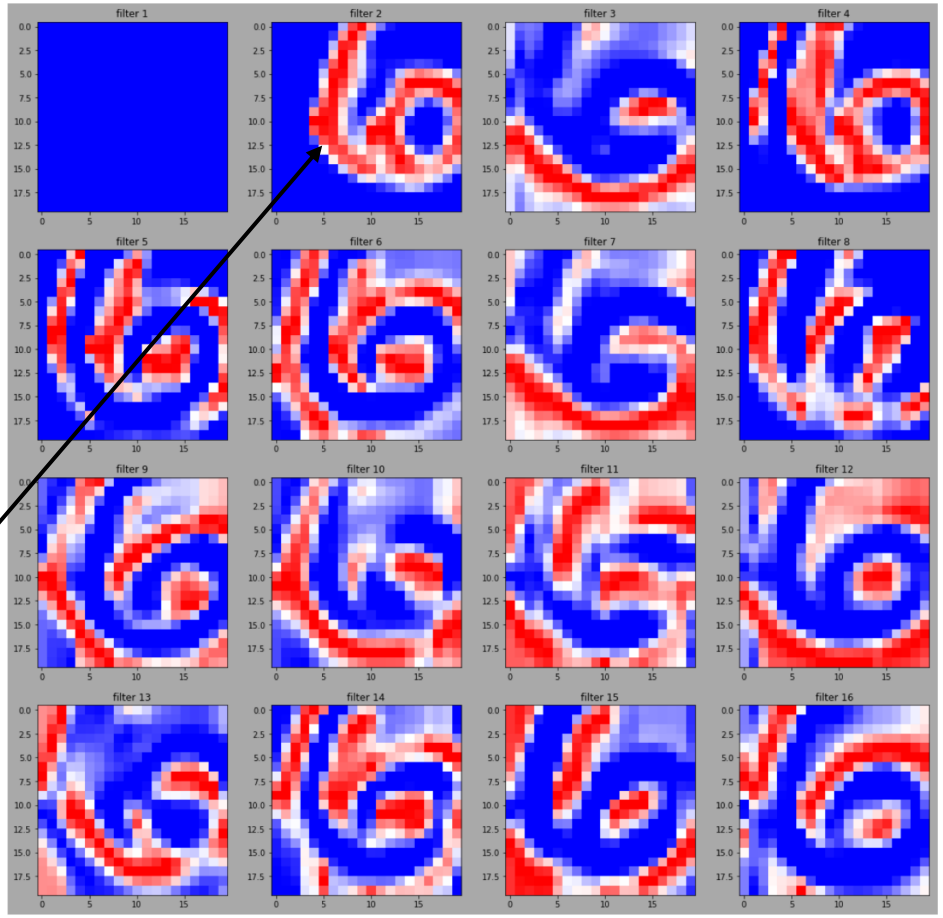
Visualizing the activations

- Most direct and effective way to understand DNNs could be to have the ability to visualize the activation layers during forward pass.
- A lot of the activations may be sparse(a lot of them zero(just blue)) for different inputs indicating dead filters because of high learning rate. Learning rate for the below mentioned examples is set to 1.0



- 3 Layered CNN with kernel size (1,1).
- Optimizer: Adadelta

- 3 Layered CNN with kernel size (3,3)
- Optimizer: Adadelta



Layer-wise Relevance Propagation (LRP)

- Decomposes the predictions of a deep neural network to pixel level relevance scores using first order approximation
- Initially proposed by Bach S. et. al (2015) https://doi.org/10.1371/journal.pone.0130140
- Computed as a backward pass using a modified gradient from the output layer to the input layer
- Implementation adopted: e-LRP (a version of LRP) as proposed by Ancona M., Ceolini E., Cengiz Ö., Gross M. (2018) in “Towards better understanding of gradient-based attribution methods for Deep Neural Networks using chain rule with a modified gradient”

$$r_i^{(l)} = \sum_j \frac{z_{ji}}{\sum_{i'}(z_{ji'} + b_j) + \epsilon \cdot sign(\sum_{i'}(z_{ji'} + b_j))} r_j^{(l+1)}$$

$$x_i * \frac{\partial^g S_c}{\partial x_i}, \quad g = \frac{f(z)}{z}$$

- **Scope of Interpretation**: Local Interpretation

Integrated Gradient

- Computes relevance score for Deep Networks for Image and Text using first order approximation
- Proposed by *Sundararajan, Mukund, Taly, Ankur & Yan, Qibi (2017)* in Axiomatic Attribution for Deep Networks
- Implementation adopted as suggested by Ancona M., Ceolini E., Cengiz Ö., Gross M. (2018) in Towards better understanding of gradient-based attribution methods for Deep Neural Networks
- Determines relevance (contribution) of an input X = $\{x_1, x_2, \ldots x_n\} \in R^n$ relative to baseline input X'
- Compute the average gradient while the input varies along a linear path from a baseline x' to x

$$IG(x) = (x_i - x_i') * \sum_{k=1}^{m} \frac{\partial}{\partial x_i} F(x' + \frac{k}{m} * (x - x')) * \frac{1}{m}$$

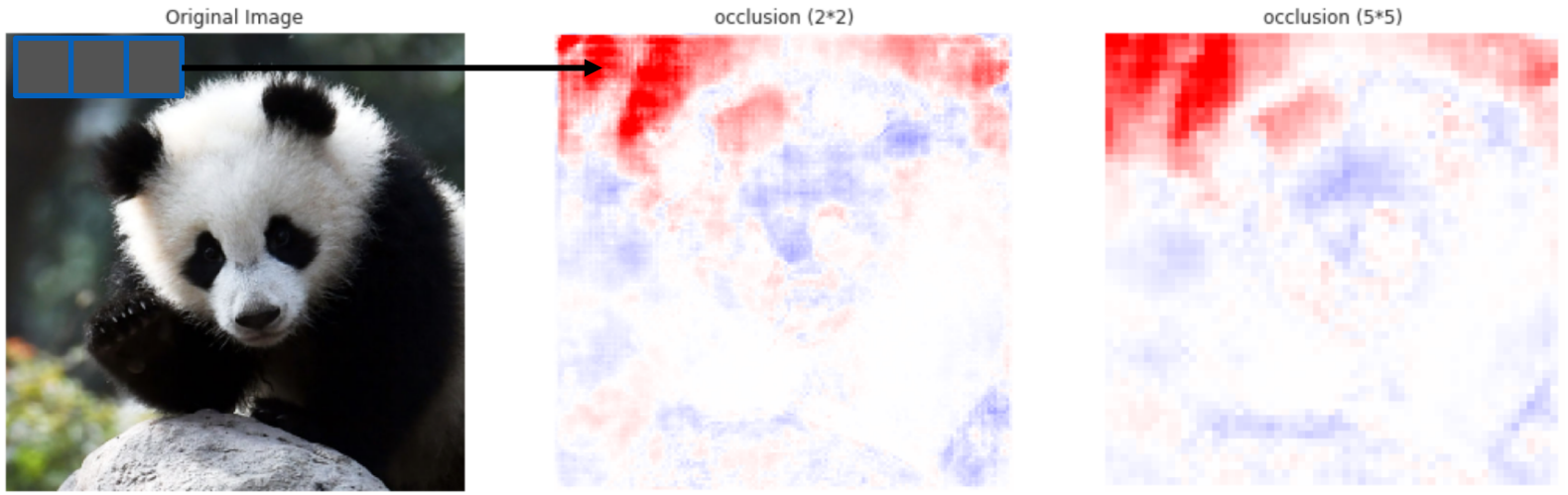
- **Baseline x':** for Image:  ; for Text: zero embedding vector
- Satisfies sensitivity and implementation Invariance
- **Scope of Interpretation**: Local Interpretation

H2O

Occlusion

- Is a perturbation based inference algorithm
- Such forms of algorithm directly computes the relevance/attribution of the input features X_i by systematically occluding different portions of the image (by removing, masking or altering them), then running a forward pass on the new input to produce a new output, and then measuring and monitoring the difference between the original output and new output.
- Helps one to compute direct estimation of the marginal effect of a feature but the inference might be computationally expensive depending on the cardinality of the feature space.
- Baseline value while perturbing through the feature space is set to 0(image space is not activated) as explained in detail by Zeiler & Fergus, 2014

Kernel size for occluding could be controlled – e.g. (1,1); (2, 2), (5, 5)



Moving beyond LIME

- Optimizes the objective function $\xi(x) = \text{argmin}_{g \in G} \ \mathcal{L}(f, g, \Pi_x) + \Omega(g)$
- Can give an indication of its own trustworthiness using fit statistics.
- Can fail, particularly in the presence of extreme nonlinearity or high-degree interactions.
- Is difficult to deploy, but there are highly deployable variants, e.g. H2O's K-LIME.
- Reason codes are offsets from a local intercept.
  - Note that the intercept in LIME can account for the most important local phenomena.
  - Generated LIME samples can contain large proportions of out-of-range data that can lead to unrealistically high or low intercept values.
- Try discretizing input features or capturing non-linear interaction by constructing features manually.
- Use cross-validation to construct standard deviations or even confidence intervals for reason code values.

Shapley Scores

- Explanation idea is borrowed from coalitional game theory
- Initial idea was proposed by Lloyd S. Shapley (1953)

  -- Lloyd S. Shapley, Alvin E. Roth, et al. *The Shapley value: Essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988. URL: http://www.library.fa.ru/files/Roth2.pdf.
- **Idea:**
  - Feature's contribution is computed by the difference between model's initial predictions
  - Followed by it's average prediction post perturbing the feature space
  - One has to be careful with features which are computationally dependent while perturbing
- Defined as $$Sh_i(v) = \sum_{S \subseteq N \setminus \{i\}, s=|S|} \frac{(n-s-1)!s!}{n!}(v(S \cup \{i\}) - v(S)), \quad i = 1, ..., n.$$

  -- Erik Strumbelj and Igor Kononenko. An Efficient Explanation of Individual Classifications using Game Theory. *Journal of Machine Learning Research*, 11(Jan):1–18, 2010. URL: http://www.jmlr.org/papers/volume11/strumbelj10a/strumbelj10a.pdf.

  -- Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. URL: http://papers.nips.cc/paper/ 7062-a-unified-approach-to-interpreting-model-predictions.pdf.
- **Scope of Interpretation**: Helps in computing Globally and locally faithful feature importance
- **Flexible:** Can be adopted in different forms - model agnostic or model specific approximations (Tree SHAP for tree ensemble methods, Linear SHAP)

H2O

Decision Boundaries

- Decision boundaries are widely used, easily understandable 2D or 3D scatter plots

-- MA Migut, Marcel Worring, and Cor J Veenman. Visualizing multi-dimensional decision boundaries in 2d. *Data mining and knowledge discovery*, 29(1):273–295, 2015.

- Is able to show data and class membership in original feature space which is helpful in understanding feature interactions.

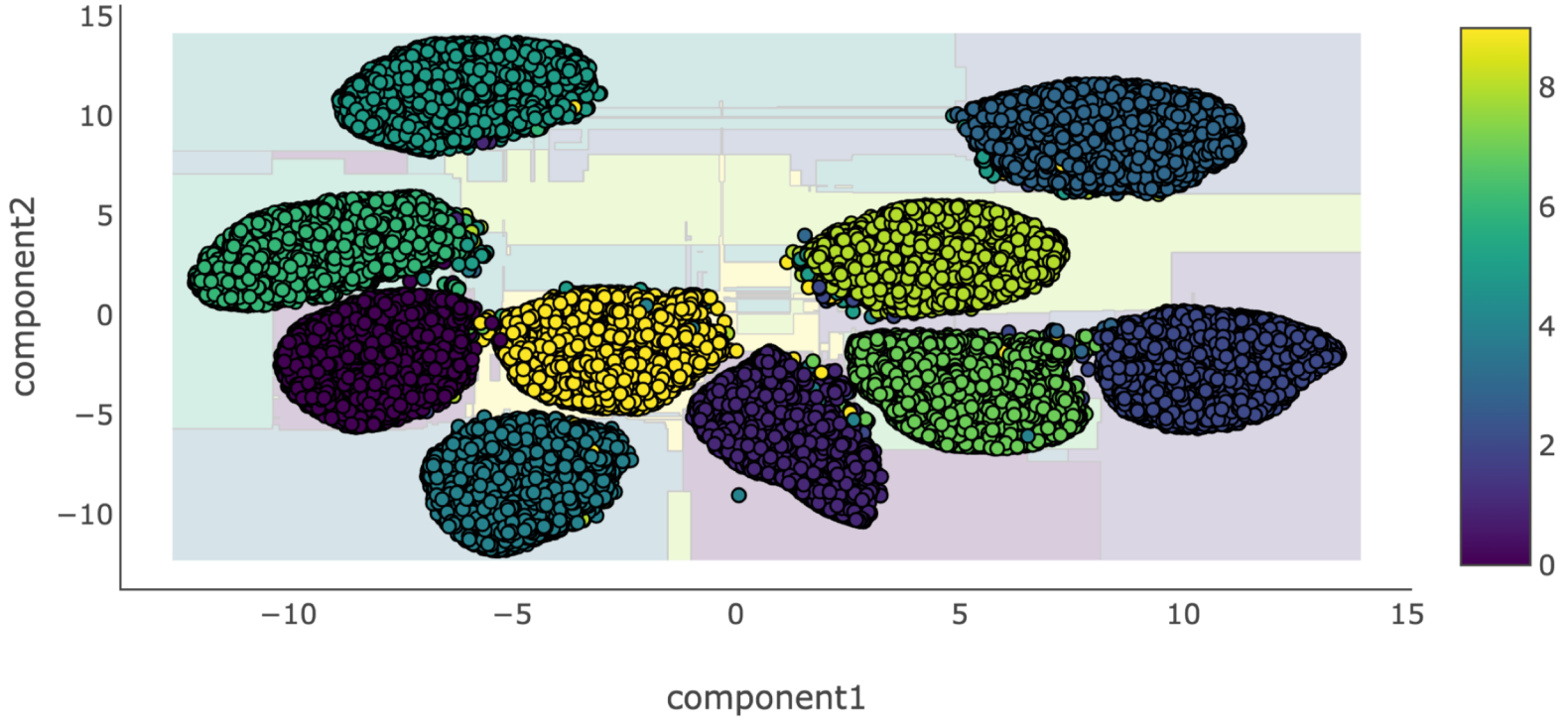


Figure: Visualizing decision boundary learned by the CovNet trained on MNIST dataset. t-SNE embedding computed using feature codes from the CNN model are used as the principal component(component1 and component2)

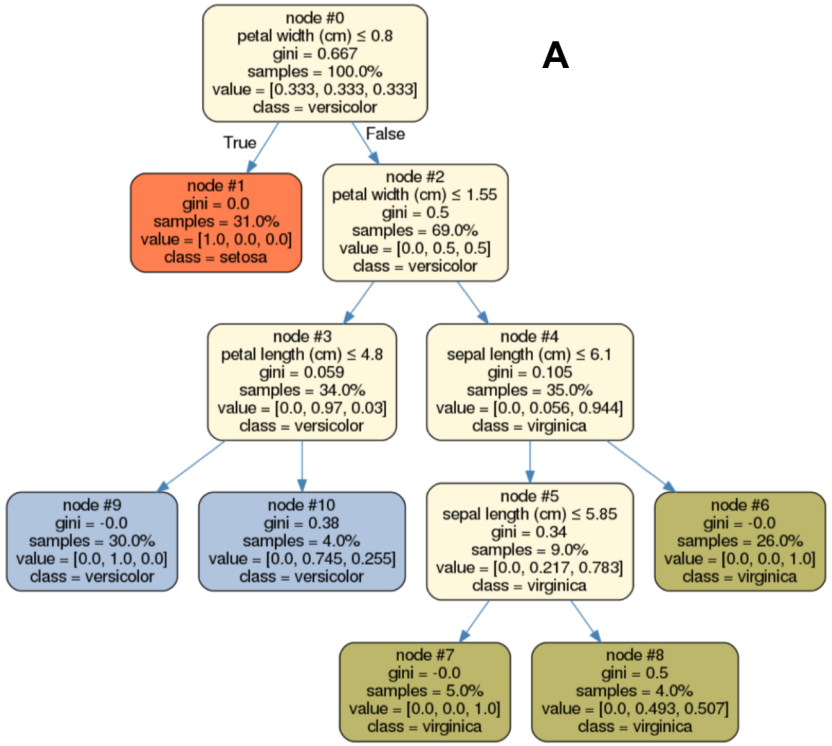
H2O

Surrogate Decision Trees (SDT)

- If the original DNN learned decision function is denoted as g and set of predictions, g(X)=Ỳ, then a surrogate tree(h_{tree}) can be learned such h_{tree} (X, Ỳ) ≅ g(X)
- The faithfulness of the decisions generated by SDT depends on how precisely the tree surrogates captures the decisions learned by the original estimator(g).

-- Craven, M., & Shavlik, J. W. (1996). Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems* (pp. 24-30).

A



B

```
if petal width (cm) <= 0.800000011920929 {
    Predicted Label: 0
} else {
  if petal width (cm) <= 1.5499999523162842 {
    if petal length (cm) <= 4.800000190734863 {
        Predicted Label: 1
    } else {
        Predicted Label: 1
    }
  } else {
    if sepal length (cm) <= 6.099999904632568 {
      if sepal length (cm) <= 5.850000381469727 {
          Predicted Label: 2
      } else {
          Predicted Label: 2
      }
    } else {
        Predicted Label: 2
    }
  }
}
```

H2O

PDP/ICE

- Helps in understanding interaction impact of two independent features in a low dimensional space visually
- Helps in understanding the average partial dependence of the target function f(Y|X)on subsetof features by marginalizing over rest of the features (*complement set of features*)

A

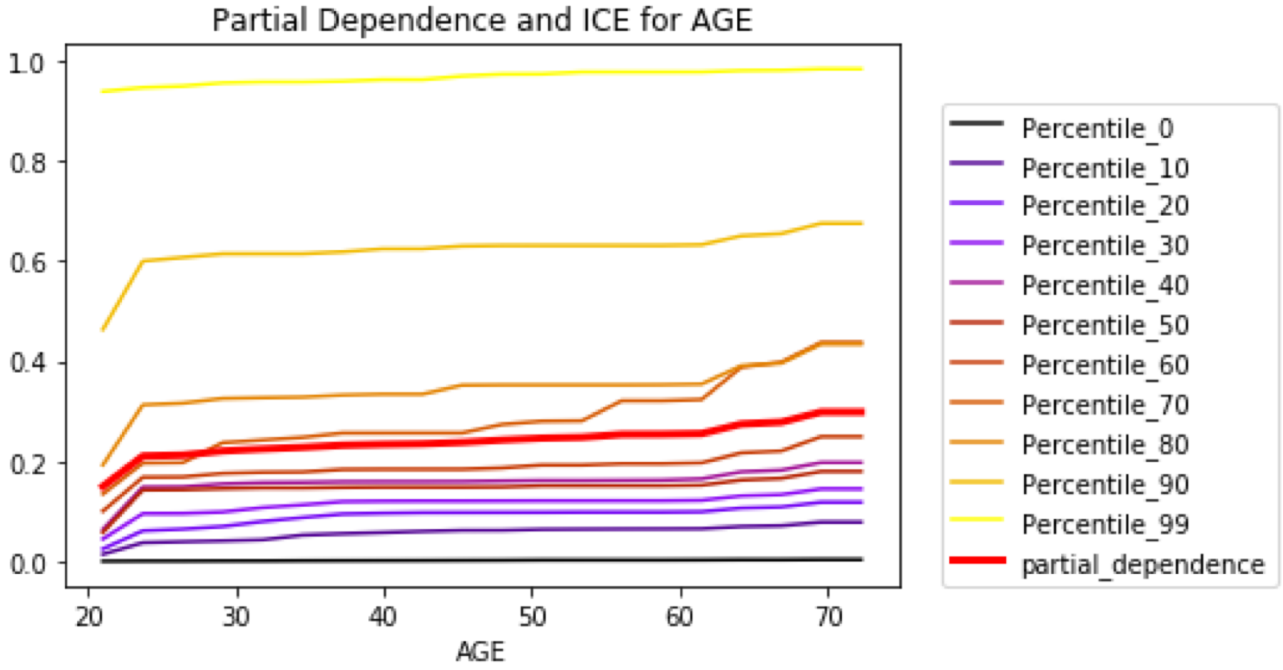
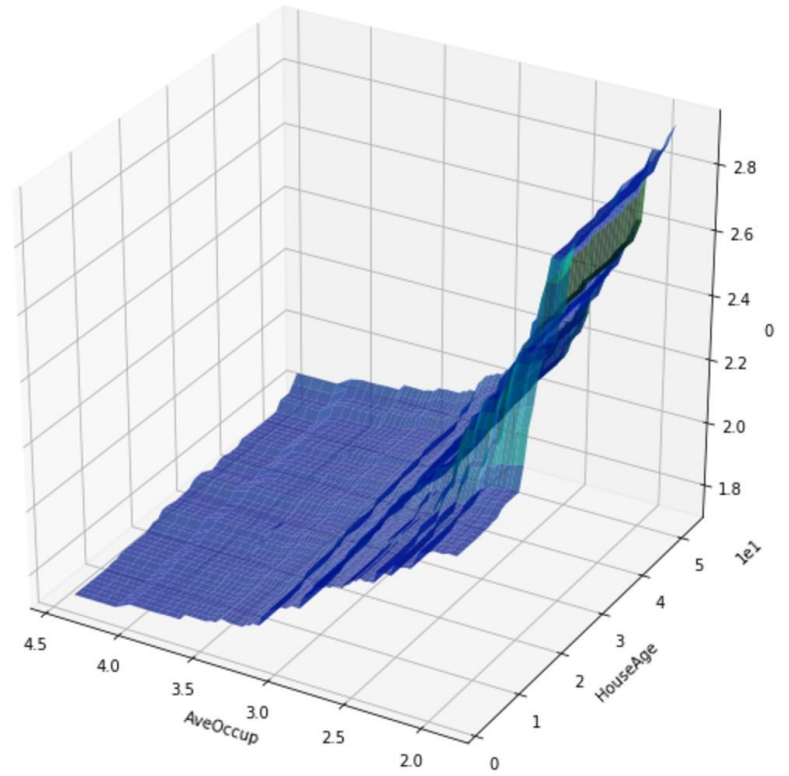


Image credit: Patrick Hall, Navdeep Gill, h2o.ai team

B



p(HouseAge, Avg. Occupants per household) vs Avg. House Value: One can observe that once the avg. occupancy > 2, houseAge does not seem to have much of an effect on the avg. house value

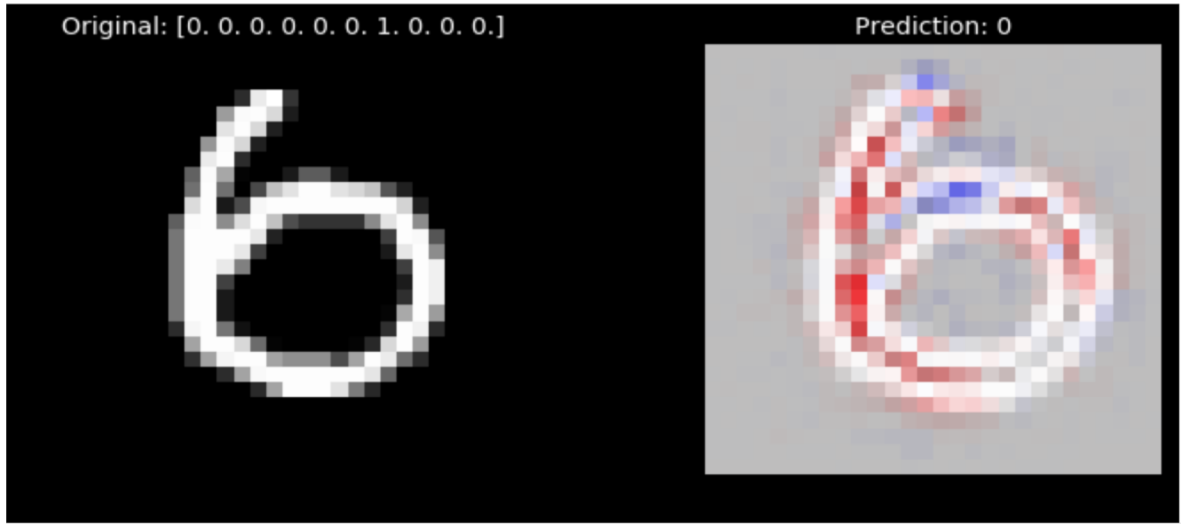
Quick Summary

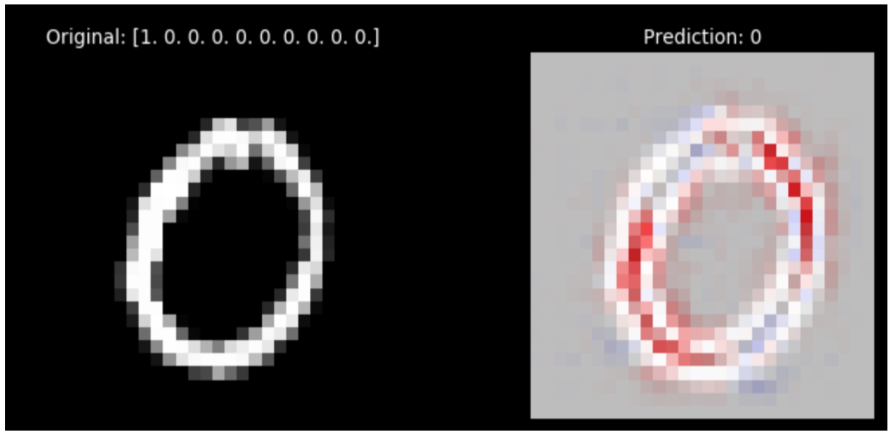
Scope of Interpretation	Algorithms
Global Interpretation	• Partial Dependence Plots(1-way and 2-way interaction) • dimensionality reduction – Principal Component Analysis, t-SNE, autoencoders, … • Visualizing feature codes(CNN) with t-SNE • Visualizing activation per layer for the complete dataset during forward pass • Surrogate Decision Tree • Decision Boundaries • Shapley Score
Local Interpretation	• Independent Conditional Expectation (ICE) • K-LIME • Shapley Scores • Surrogate Decision Tree • Decision Boundaries • Visualizing activation per layer for a single row • Layer-wise relevance propagation(e-LRP) • Integrated Gradient • Occlusion

Table: The above summarization is by no means an exhaustive list of algorithms, but just a summarization of some of the methods selected for this discussion.

Experiments

What about Model Stability?





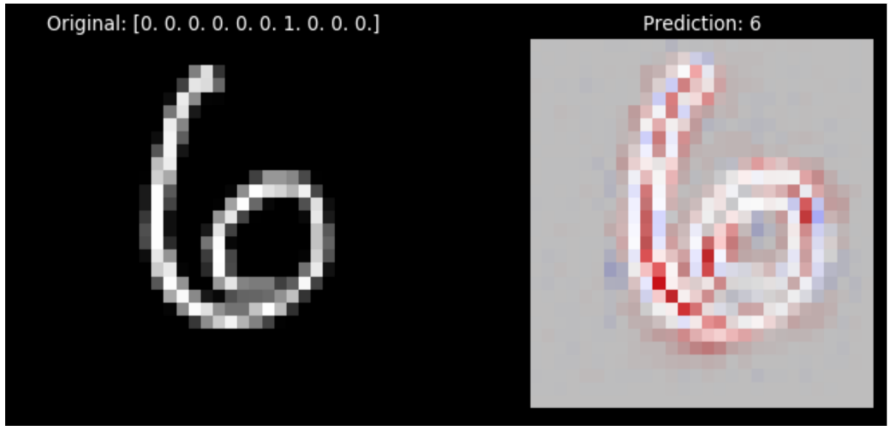
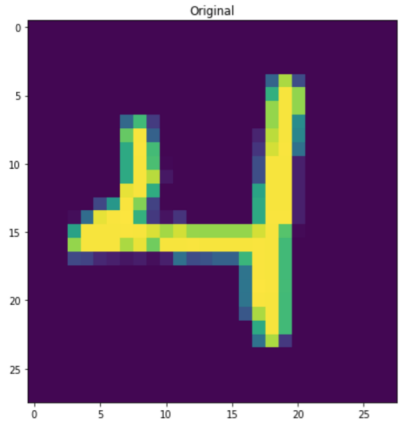
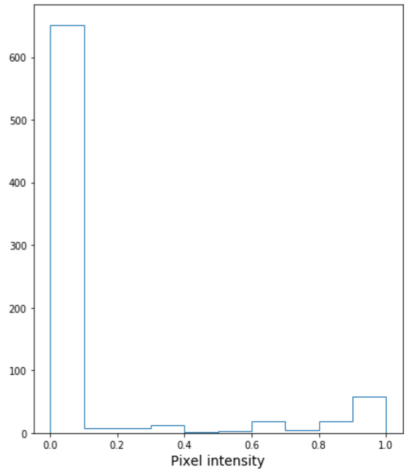


Figure: An MNIST experiment with CNN model with 98.8% train and 98.6 test 'Accuracy'. Interpretation CNN model with 'ReLU' activation using e-LRP. Image-6 on the left is in-correctly classified as 0. Skater provides the ability to infer the cause of mis-classification
(Pixels colored in Red have a positive influence and Blue negative influence). Images share a semantic properties globally. In the above example we can see 6 and 0 sharing semantic properties around the lower curvy round 'O', probably the reason for misclassification.

H2O

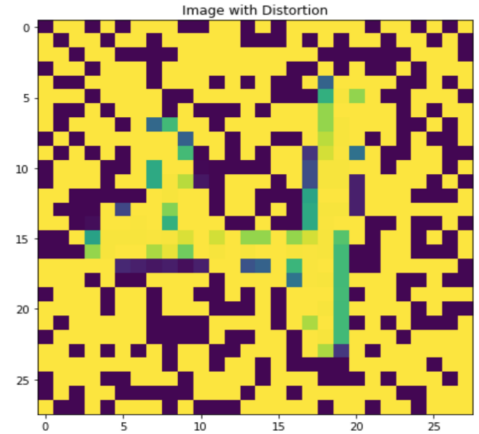
Identifying blind spots

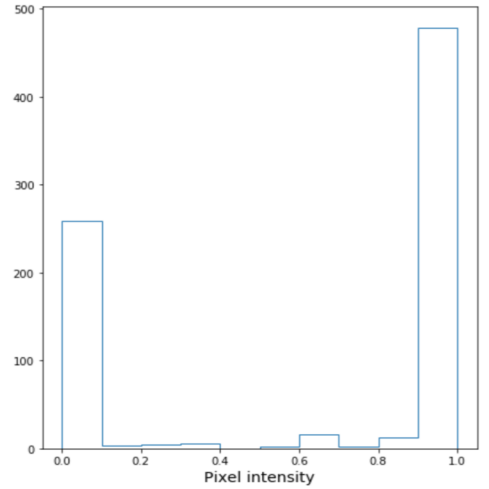


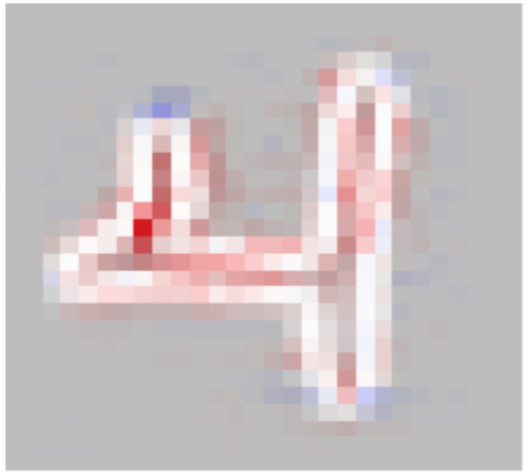


Distortion(D)

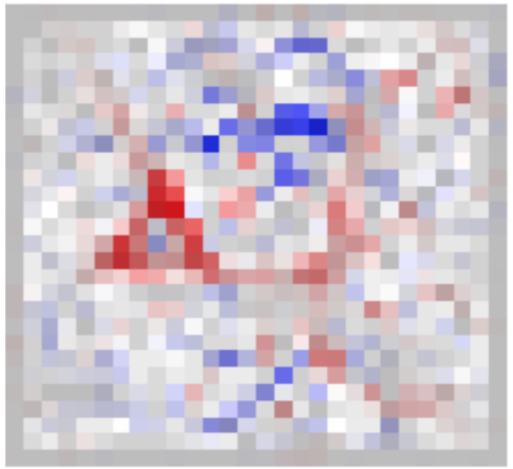
$X + r \in [0, 1]^m$



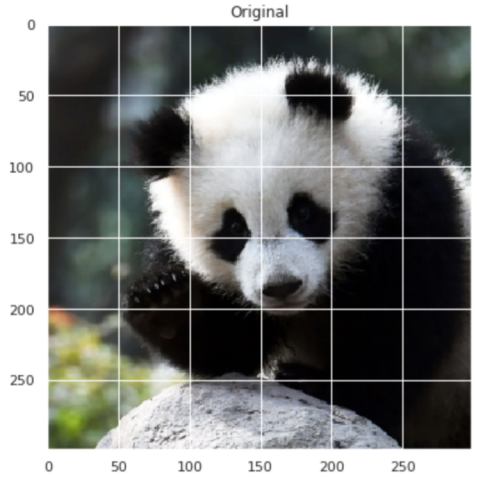


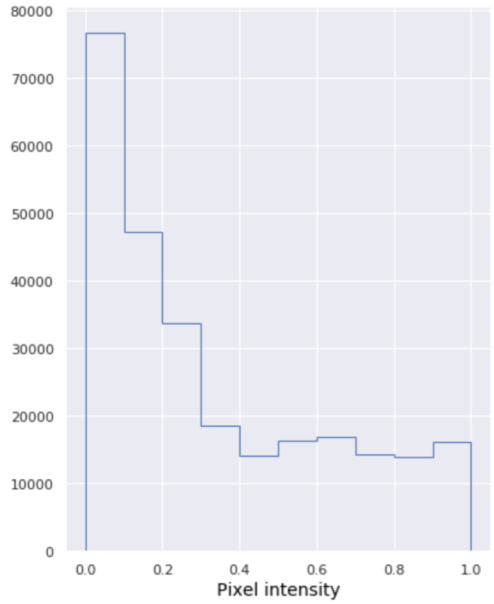


Relevant Image pixels are retained and correctly identified

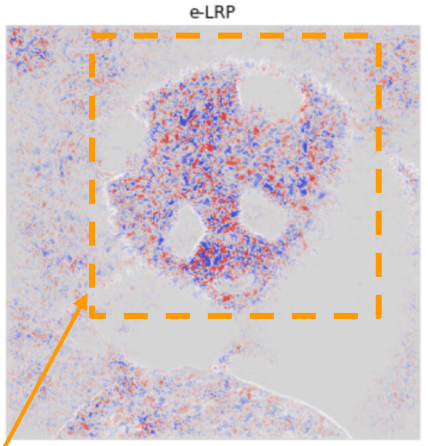


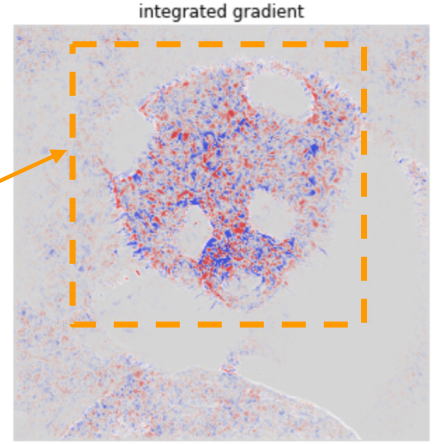
Creating Adversarial Examples



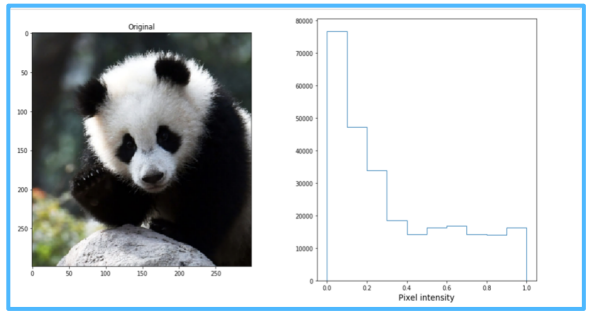


	Class Labels	Scores
0	giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca	0.934644
1	soccer ball	0.001220
2	space shuttle	0.000602

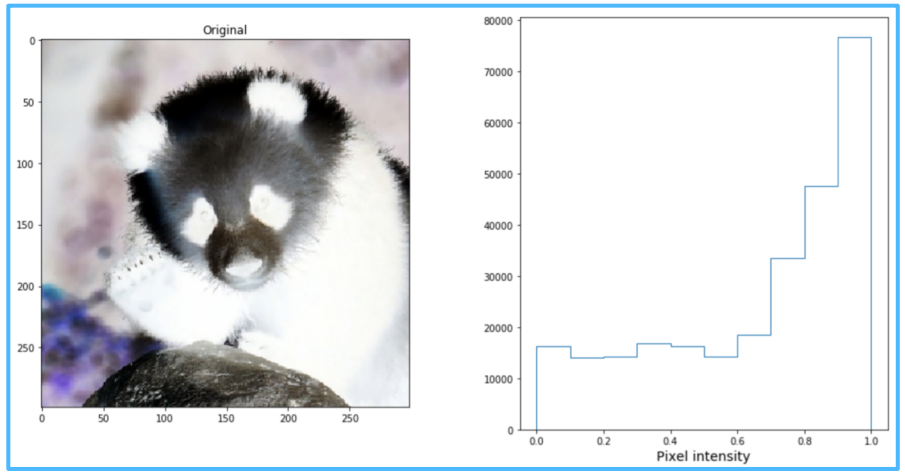




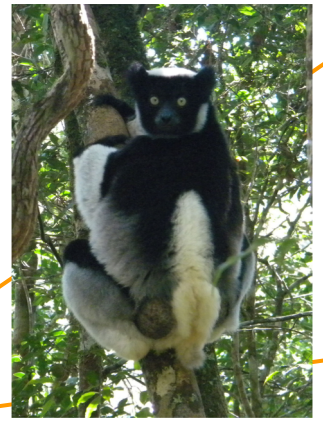
Conditional adversarial tested against pre-trained Inception-V3



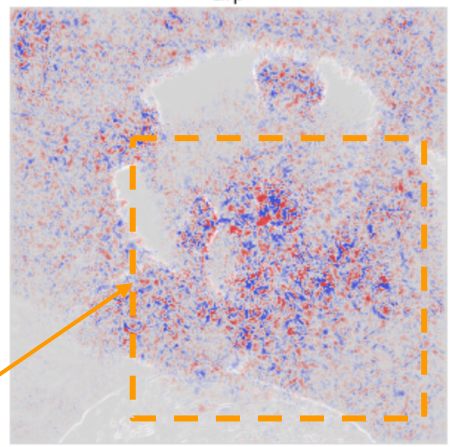
	Class Labels	Scores
0	giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca	0.934644
1	soccer ball	0.001220
2	space shuttle	0.000602



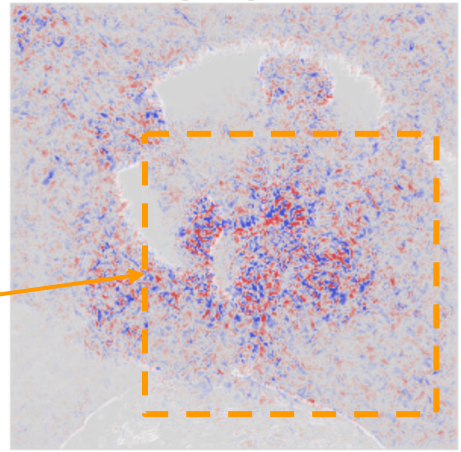
	Class Labels	Scores
0	indri, indris, Indri indri, Indri brevicaudatus	0.266485
1	guenon, guenon monkey	0.037744
2	colobus, colobus monkey	0.035104



elrp



integrated gradient

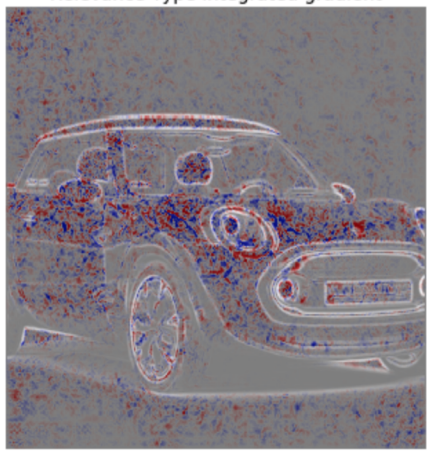


More Examples

Input Image

sports car: 0.54%

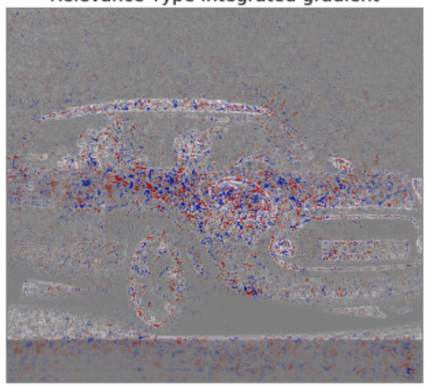
Relevance Type integrated gradient



Input Image

grille: 0.21%

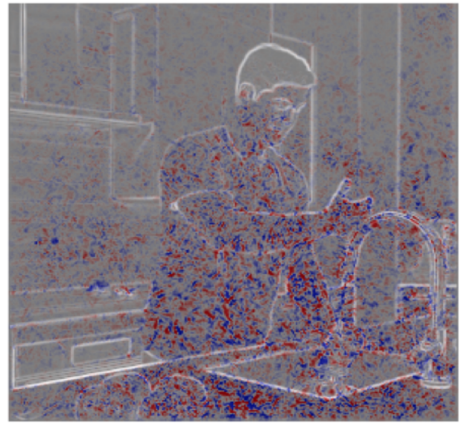
Relevance Type integrated gradient



Input Image

washbasin: 0.43%

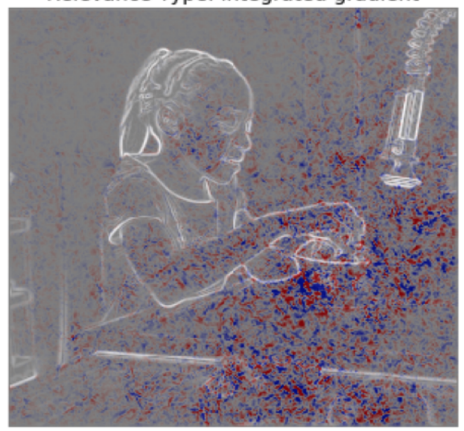
Relevance Type: integrated gradient



Input Image

ping-pong ball: 0.14%

Relevance Type: integrated gradient



Another approach: Enable Interpretability natively for DNNs

Image Segmentation/Object Detection: e.g. Mask R-CNNs (Kaiming He et al. Region-based CNN): prediction is accompanied by segmentation mask for region of interest

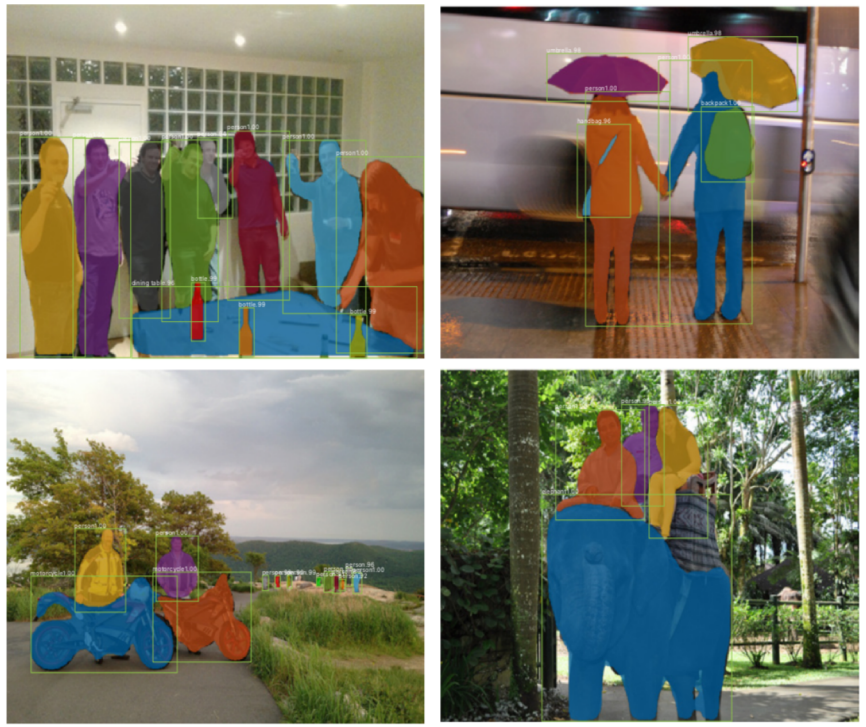


Figure:
Mask R-CNN on COCO dataset.
Reference: https://arxiv.org/abs/1703.06870

What about text?

X = excellent tale of two boys that do whatever they can to get away from there abusive father lord of the rings star elijah wood is outstanding in this unforgettable role this movie is one of the main reasons I haven't touched a single beer and never will as long as i live that might make me sound like a nerd but that's what I have to say it is a wonder why this isn't as a classic american tale

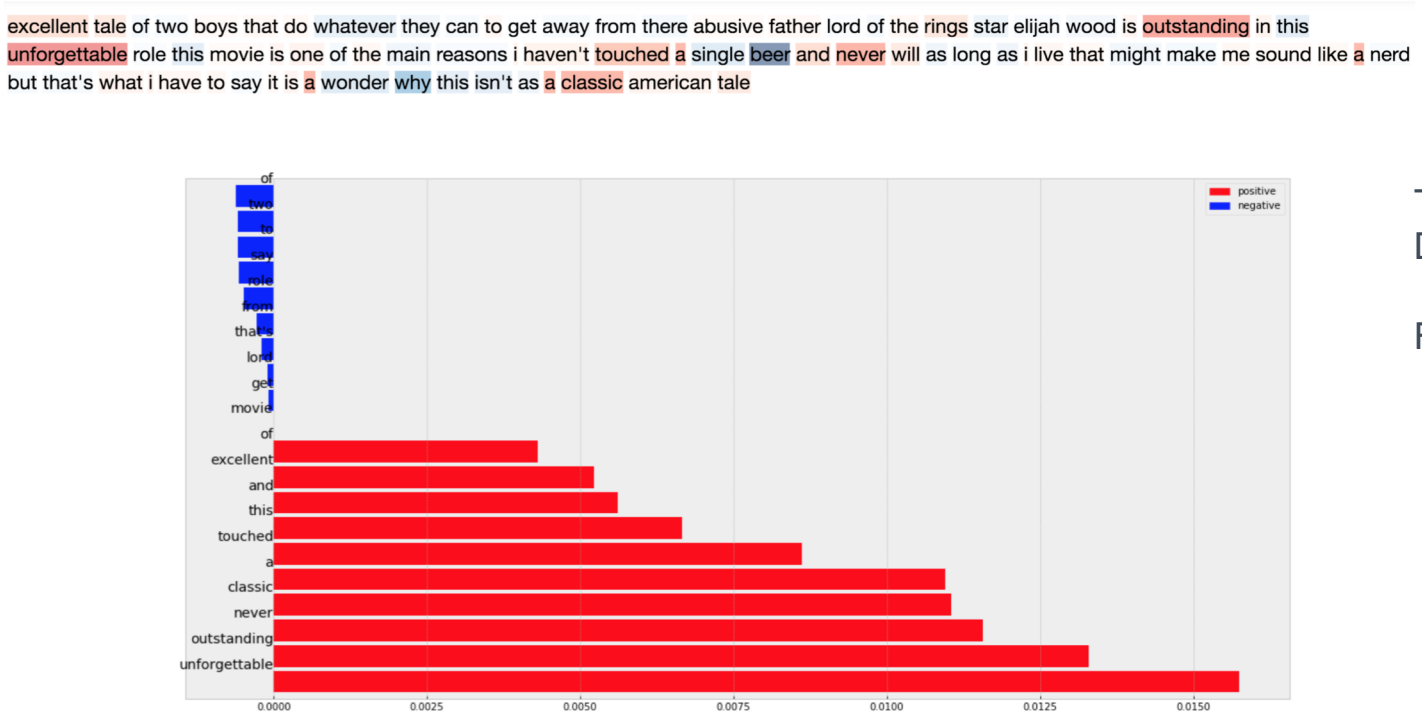


Figure: Sentiment Analysis use-case using IMDB dataset. Above plots illustrates features positively and negatively influencing prediction

Trained Model: CNN for sentiment analysis
Dataset: IMDB

For a trained DNN classifier model(F),

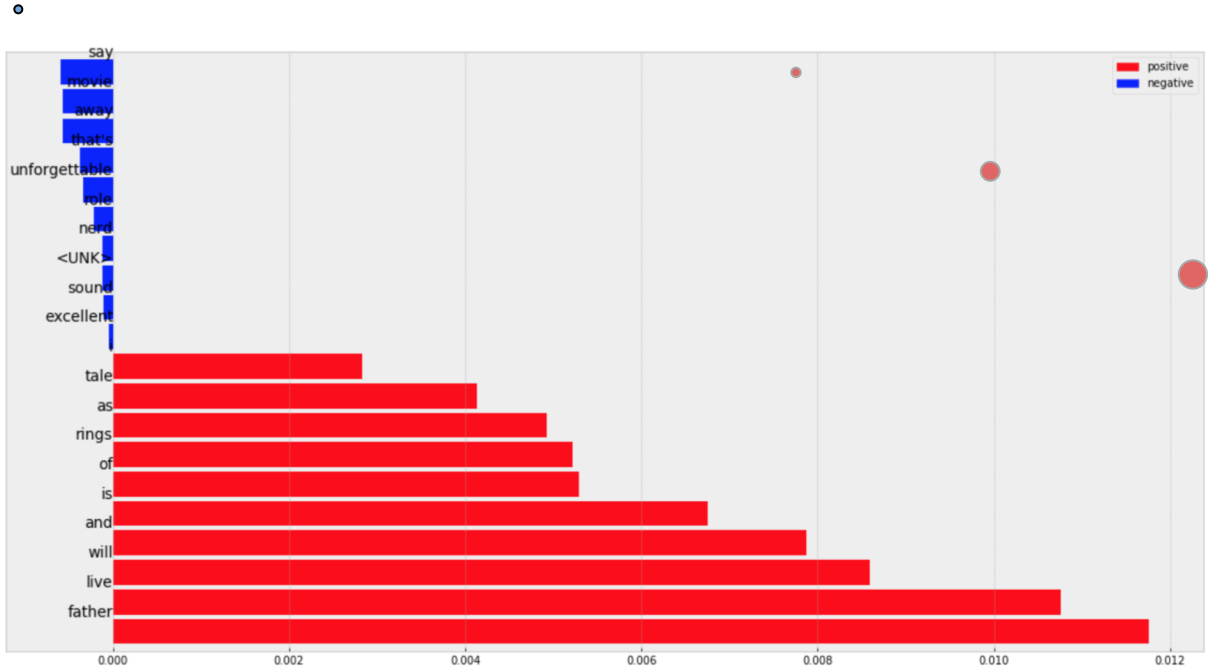
- Craft an adversarial attack by adding perturbation ΔX
  $X^{adv} = X + \Delta X$
- ΔX = <Insert, delete, replace>
- $F(X) \neq F(X^{adv})$

H2O

Pixel perturbation == <insert, delete, replace> for text

<UNK> tale of two boys that do whatever they can to get away from there abusive father lord of the rings star elijah wood is <UNK> in this <UNK> role this movie is one of the main reasons i haven't <UNK> a single beer and <UNK> will as long as i live that might make me sound like a nerd but that's what i have to say it is a wonder why this isn't as a <UNK> american tale

excellent tale of two boys that do whatever they can to get away from there abusive father lord of the rings star elijah wood is in this unforgettable role this movie is one of the main reasons i haven't touched a single beer and never will as long as i live that might make me sound like a nerd but that's what i have to say it is a wonder why this isn't as a classic american tale



Negative signals

positive signals

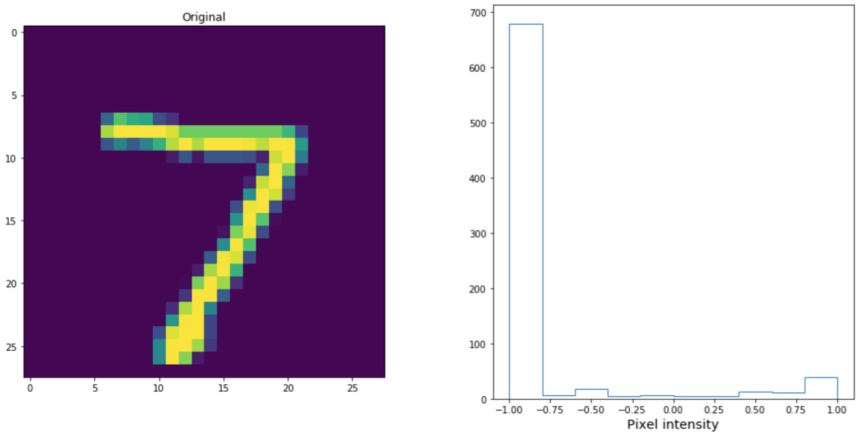
Adversarial Training(Counterfactual Examples)

Inputs modified maliciously to yield erroneous outputs which may appear unmodified to humans

Adversarial Generation

- How can we craft adversarial examples easily, i.e. examples to fool a predictive model with high confidence?

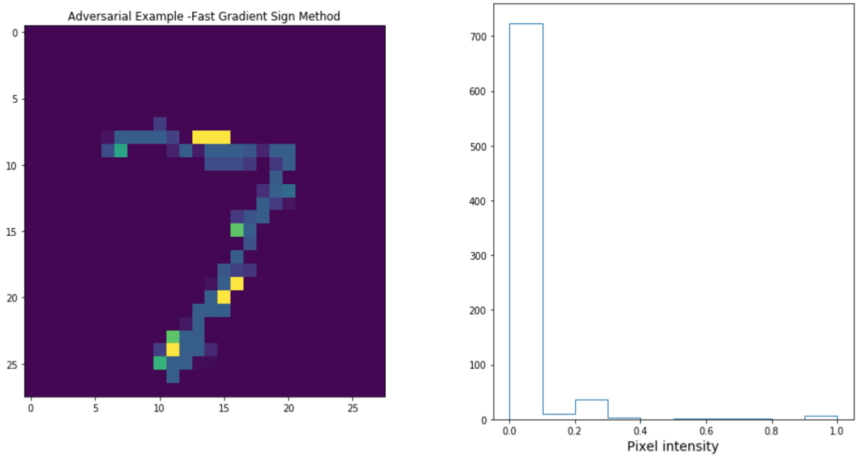
Predicted Label: 7



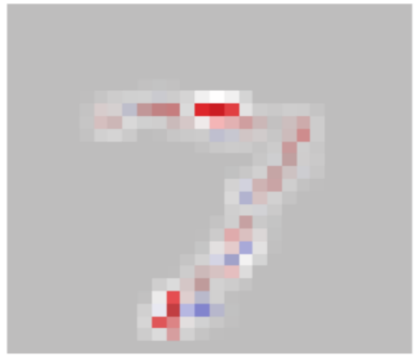
Distortion(D)

Using FGSM

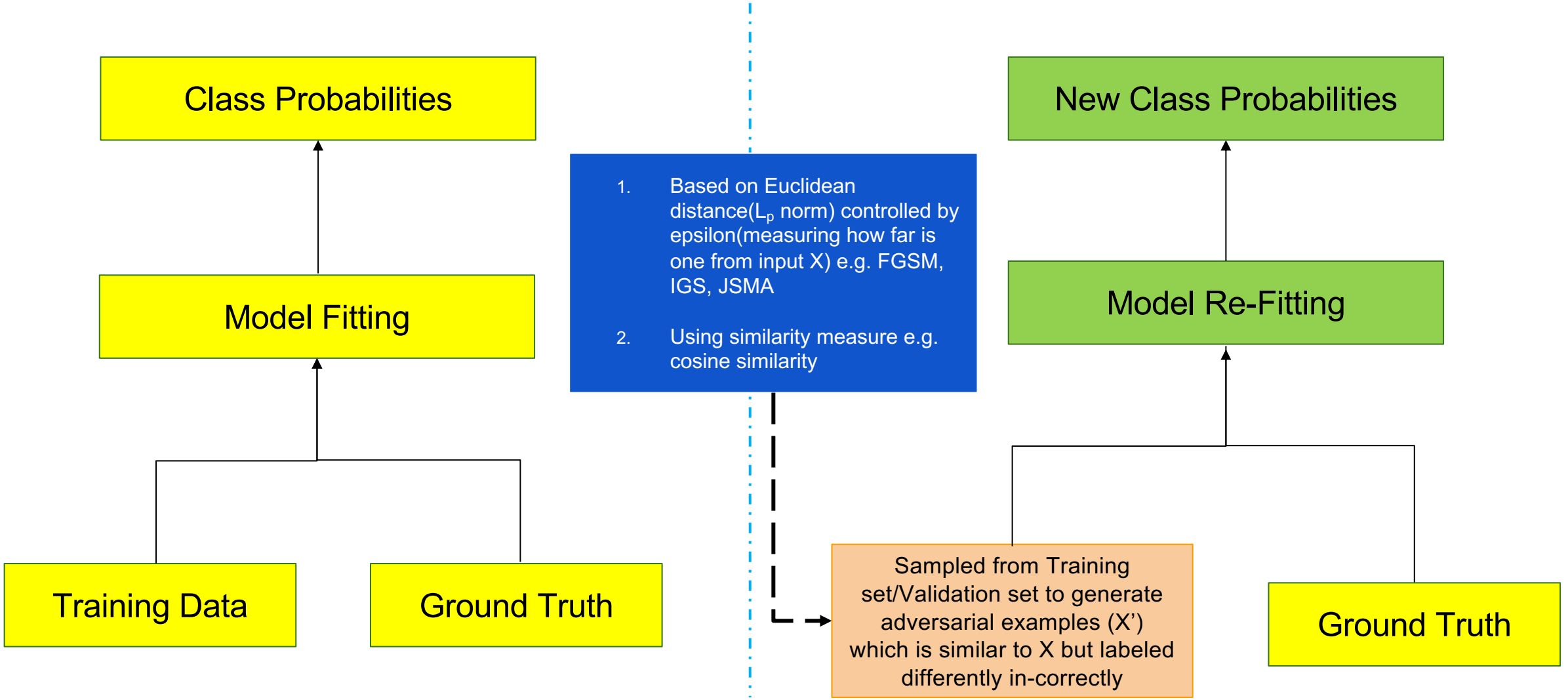
Predicted Label: 3



Inference using Occlusion



Automated Adversarial Generation



-- Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016, May). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 582-597). IEEE.

H$_2$O

Early benchmark

Fit a Model	Before Adversarial Attack		
Apply adversarial attack on the test set and evaluate	Train Set	Validation/Holdout Set	Test Set
	0.99	0.98	0.98
			0.83
Re-Fit a Model			
Adverarial Attack on validation set using same parameter to generate examples as used for testset earlier and re-train	Train Set	Validation/Holdout Set and evaluation without re-training	Test set
	0.99	0.89	0.83
		Re-trained using new validation set as part of training	
		0.99	0.99

Figure: Adversarial attack on MNIST dataset, using FGSM.
More exploration is pending.

Summary

- **Misconception:** Model Interpretation means explanations generated using surrogate model. That's not true.
- Accurate explanation could be generated using the original model(base estimator)
- Many other forms of explanations are available with well founded mathematical proof . However, selection of right algorithm could be contextually dependent
- Helps in understanding the evolution of the features during training time, may result in further insights depending on the domain
- Helps in diagnose potential problems in the model – identifying context driven decisions
- Helps reduce trial and error while building optimizing the model for generalization (balancing between bias and variance)
- Helps in selecting better DNN architecture resulting in better optimized classification performance.
- On-going effort:
  - More research is needed for evaluating different aspects of interpretability.
  - Making the algorithms scale efficiently with data

References

- https://github.com/h2oai/mli-resources
- Skater(library is going through changes): https://github.com/datascienceinc/Skater
- t-SNE: https://www.cs.tau.ac.il/~rshamir/abdbm/scribe/17/lec05.pdf
- Andrej Karpathy: http://cs231n.github.io/understanding-cnn/
- Paul, M. J. (2016). Interpretable machine learning: Lessons from topic modeling. In *CHI Workshop on Human-Centered Machine Learning*.
- Gu, S., & Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140. PloS one, 10(7):e0130140
- Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.
- Liang, B., Li, H., Su, M., Bian, P., Li, X., & Shi, W. (2017). Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

References

- Arras, L., Horn, F., Montavon, G., Müller, K. R., & Samek, W. (2017). " What is relevant in a text document?": An interpretable machine learning approach. *PloS one*, *12*(8), e0181142.

- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

- Ancona, M., Ceolini, E., Oztireli, C., & Gross, M. (2018). Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations (ICLR 2018)*.

- Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., & Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*, *3*(3), e10.

- Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. In *Advances in neural information processing systems* (pp. 5574-5584).

- Freitas, A. A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, *15*(1), 1-10.

QnA

@MaverickPramit

@h2oai

https://www.h2o.ai/careers/

https://github.com/h2oai/mli-resources

https://github.com/pramitchoudhary/ODSC-west