

Interpretability in machine learning models by Local Interpretable Model-Agnostic Explanations (LIME)

CS542 Final Project

Advisor: Dr. Peter Chin

TAs: Andrew Wood, Ryan Yu

Team 10

Members:

Bingquan Cai bqcai@bu.edu

Chuwei Chen chenchuw@bu.edu

Xiaowei Ge xwge@bu.edu



Chuwei Chen contributed to SVM-LIME part; Bingquan Cai contributed to the Random Forest-LIME part; Xiaowei Ge contributed to the CNN-LIME part. All participated in the concept and model establishment.

Interpretability in machine learning models

Interpretability

Understanding how machine learning is making decisions

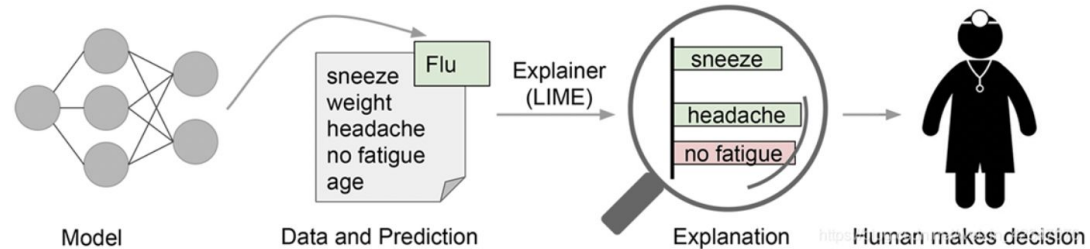
- Decision trees
- Linear classification
- Neural networks

Importance statement

- Could make the decisions more trustworthy
- Crucial in the fields (medical, autonomous driving, etc) which used to highly rely on human experience

Major challenge

The trade-off between model complexity and interpretability



Interpretability in machine learning models

Interpretability

Understanding how machine learning is making decisions

- Decision trees
- Linear classification
- Neural networks

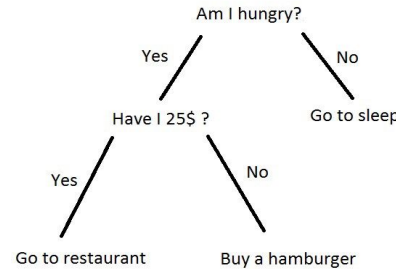
Importance statement

- Could make the decisions more trustworthy
- Crucial in the fields (medical, autonomous driving, etc) which used to highly rely on human experience

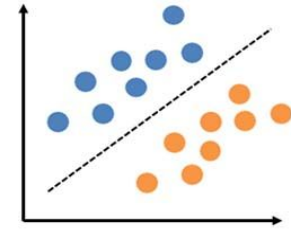
Major challenge

The trade-off between model complexity and interpretability

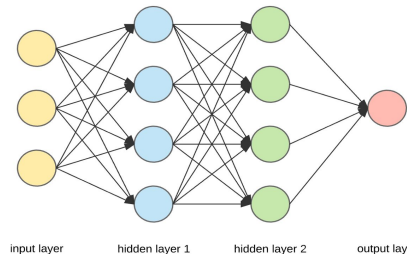
Decision trees



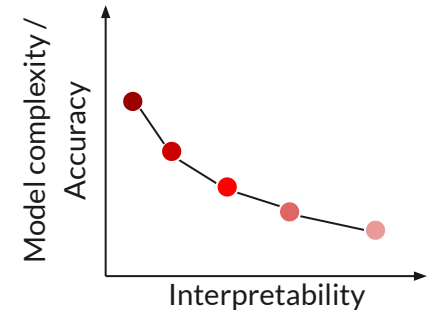
Linear classification



Neural networks



Challenges



Interpretability in machine learning models

Can we achieve high interpretability while maintain the model complexity to achieve ★ ?

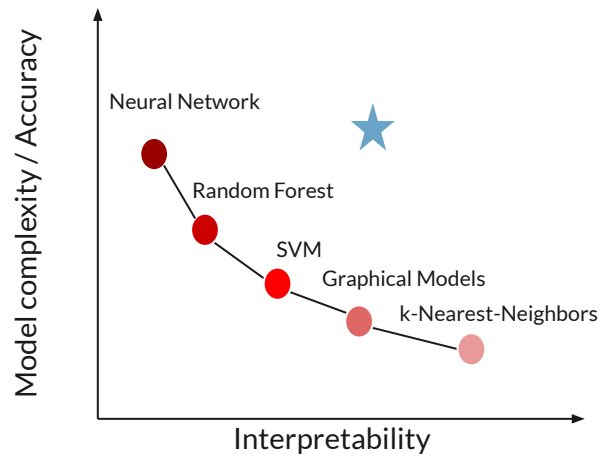
Method:

Local Interpretable Model-agnostic Explanations (LIME)

Local: only try to interpret with local perturbation

Interpretable: human interpretable features

Model-agnostic: applicable to general machine learning models



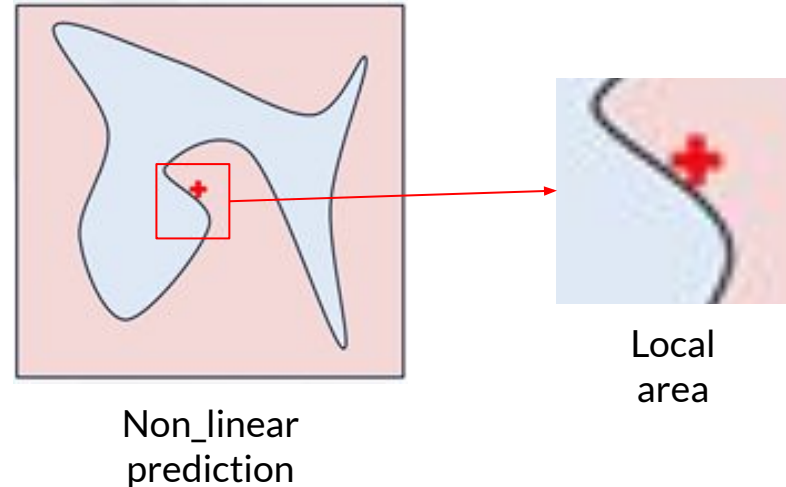
L: locally I: human interpretable M: treat model as a black box E: explanation

How LIME works

- Data set \rightarrow black box ml algorithm \rightarrow training result
- Decision boundary non-linear
- Prediction of new data (why)
- Hard to summarize whole / one explanation

LIME:

- Zoom into local area
- Find local linear model
- Simple explanation



How LIME works

Input data

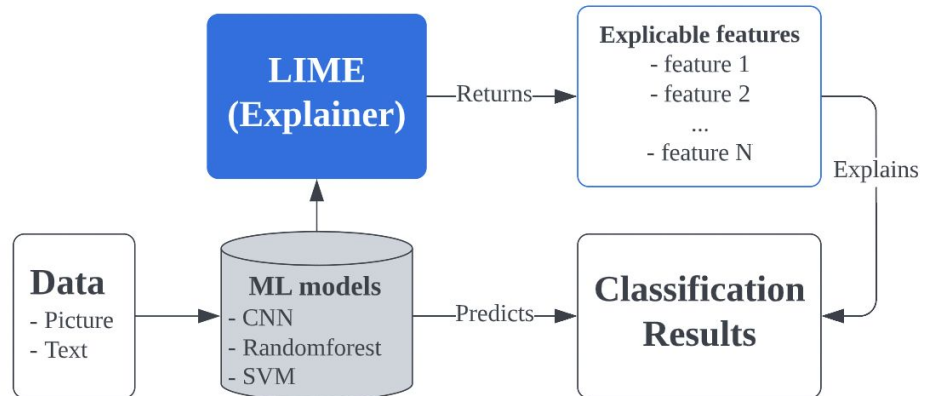
$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Good approximation

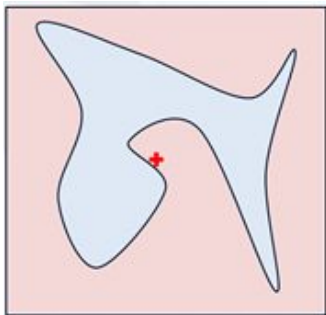
Stay simple

Complex model Simple model Proximity

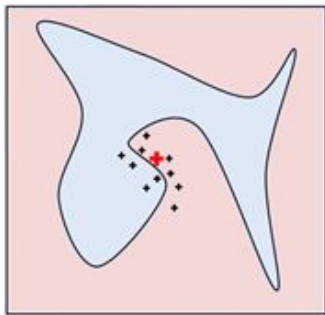
- Local approximation
- Regularize the complexity



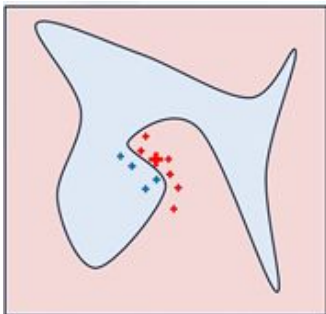
How LIME works



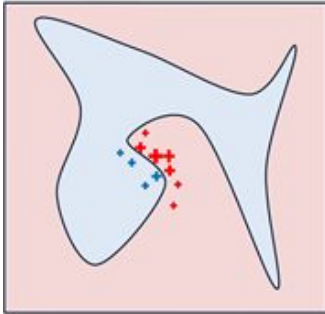
Input data



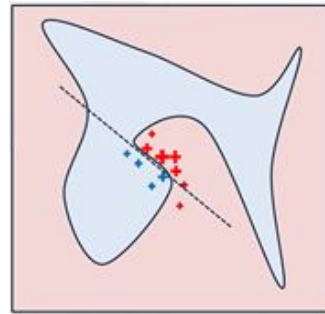
New data



Make prediction



Weighted



Local linear model

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

Loss function

- New data set
- Fit linear classifier
- Proximity (weight the loss)
- Omega (sparse linear model)

Problem statement

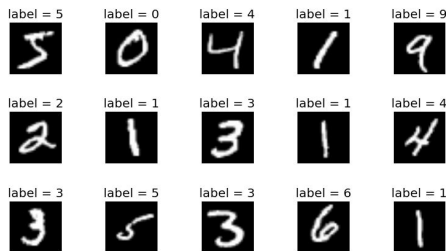
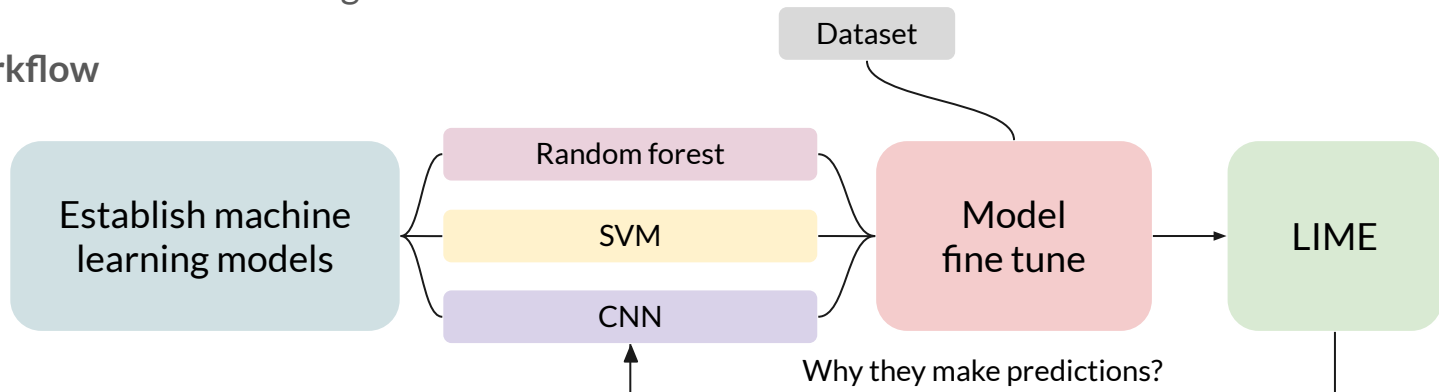
Dataset

1. Comments classifier
2. MNIST
3. ImageNet

Goal

Evaluate LIME explanation results on different machine learning models

Workflow



SVM with LIME - Comments Classification

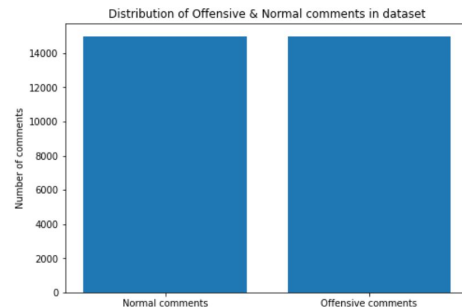
- Normal comments

id	comment_text
0000997932d777bf	Explanation\nWhy the edits made under my usern...
000103f0d9cfb60f	D'aww! He matches this background colour I'm s...
000113f07ec002fd	Hey man, I'm really not trying to edit war. It...
0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...
0001d958c54c6e35	You, sir, are my hero. Any chance you remember...

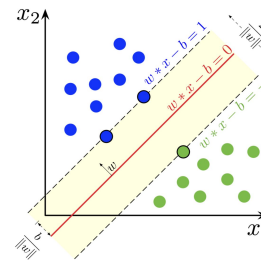
- Offensive comments

id	comment_text
414340cb8e17e3cd	"\n\nYOU MAKE ME WANT TO PUKE! THE WAY YOU ACT...
41fe17d86e765e7b	WIKI NAZI!! \n\nThat's all you are, you even re...
420754a82f6749a9	oh shit you gon get banned
421010b18bfd1899	Just To Let You Know. You have no life.\nI hat...
425ec3c32af1b68a	"\n\n good job for sucking dick \n\n dick tro...

- Distribution of dataset



- SVM classifier w/ ~90% accuracy (linear kernel)

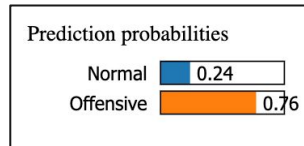


SVM with LIME - Comments Classification

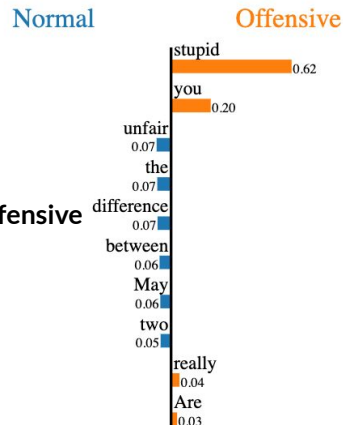
- An offensive comment:

“Are you really that stupid? Do you know the difference between these two words: unfair and allege? May I suggest going back to nursery school and learn basic English, again.”

- LIME result w/ SVM classifier:



❑ Offensive comment -> Predicted as Offensive

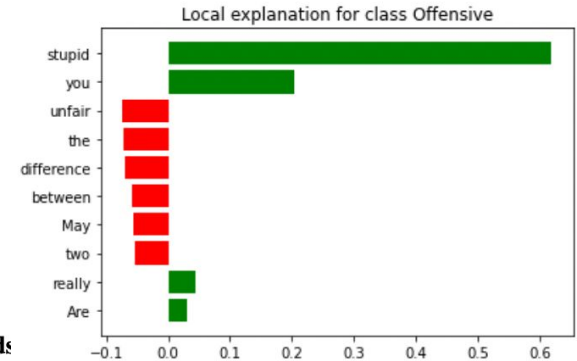


Text with highlighted words

Are you really that **stupid**?

Do you know the difference between these two words: unfair and allege?
May I suggest going back to nursery school and learn basic English, again.

- Local explanation:

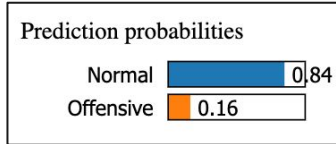


SVM with LIME - Comments Classification

- A normal comment:

“Perfection, my only wish is, that this were a live stream.”

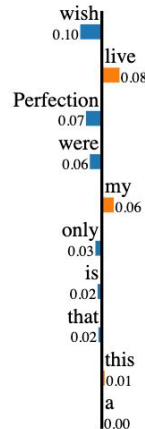
- LIME result w/ SVM classifier:



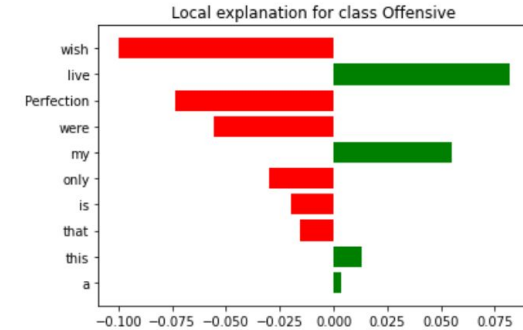
☐ Normal comment -> Predicted as Normal

Normal

Offensive



- Local explanation:



Text with highlighted words

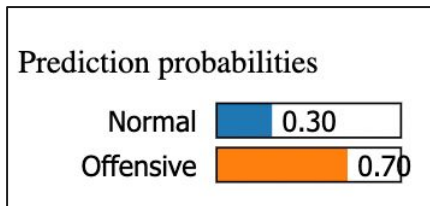
Perfection, my only wish is, that this were a live stream

SVM with LIME - Comments Classification

- A normal comment:

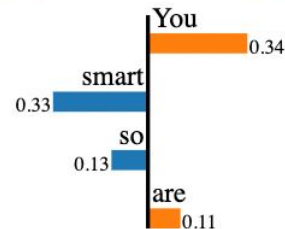
"You are so smart!"

- LIME result w/ SVM classifier:



Normal

Offensive



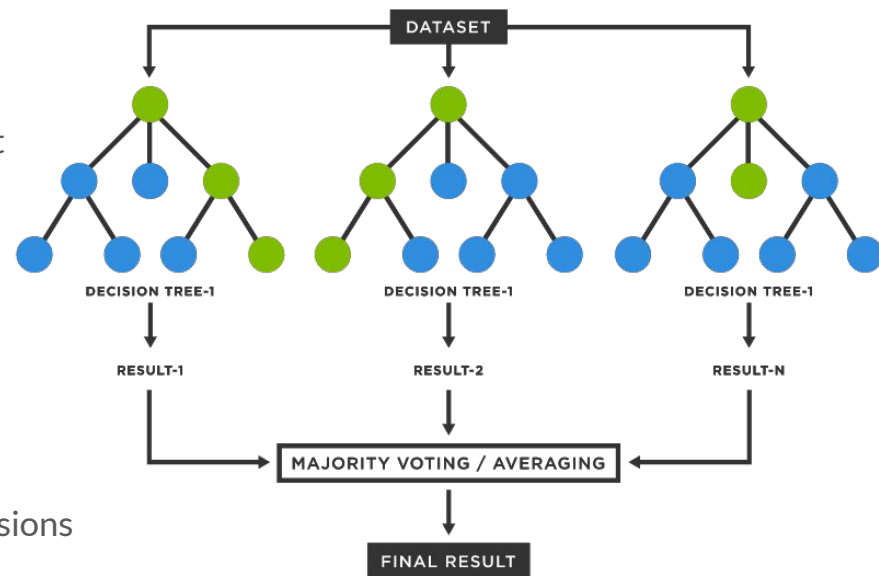
Text with highlighted words

You are so smart!

- ❑ Normal comment -> **Wrongly predicted as Offensive.**
- ❑ Why?

Random forest with LIME

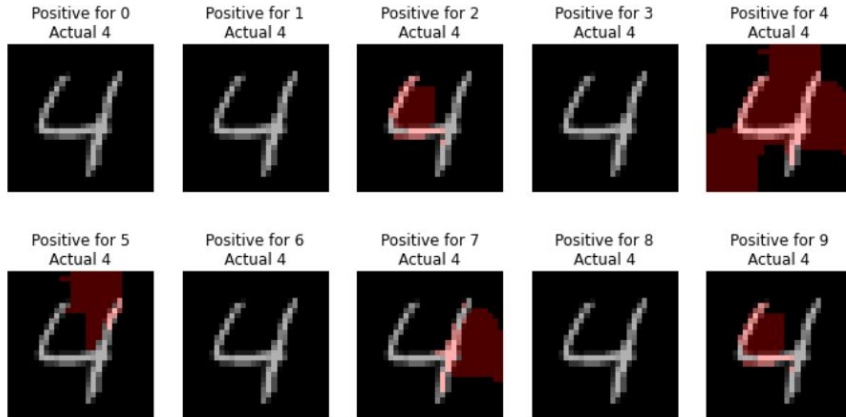
- How random forest work?
 - Select random samples from a given dataset
 - Construct a decision tree for each sample
 - Get prediction from each decision tree
 - Vote for each predicted result
 - Final prediction
- Feature importance
 - Gini importance
 - Used for selecting variable and making decisions
- Hard to understand...



Random forest with LIME

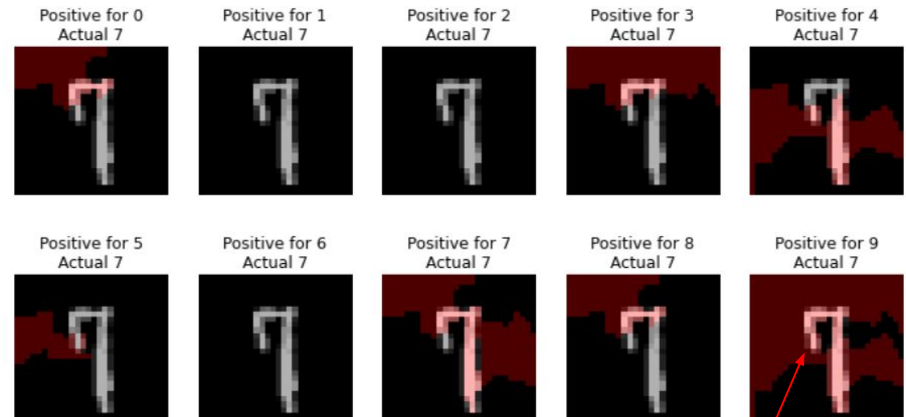
Explanation for correct prediction

Number: 4 Prediction: 4

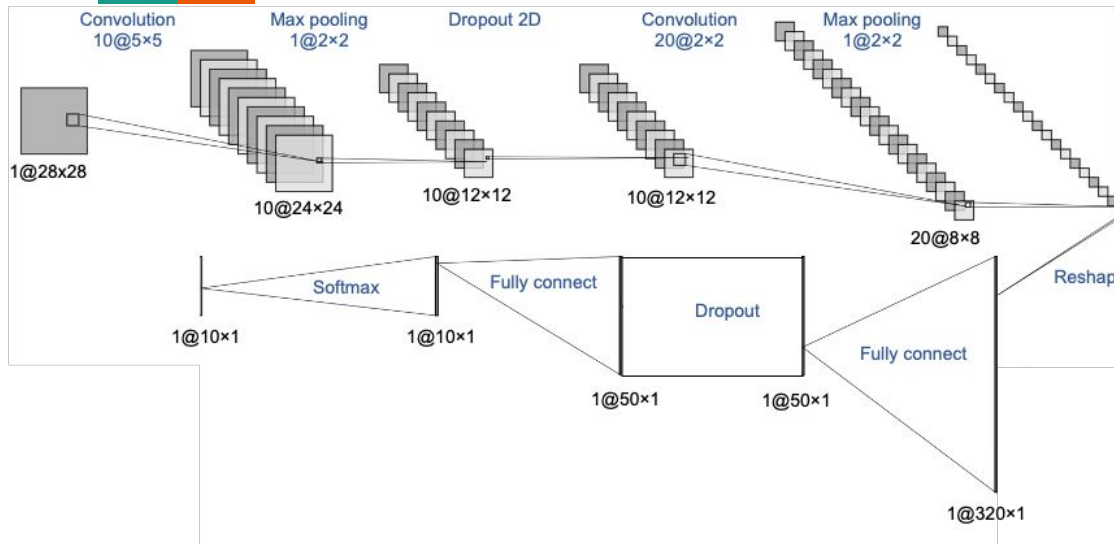


Explanation for wrong prediction

Number: 7 Prediction: 9



CNN architecture and fine tune

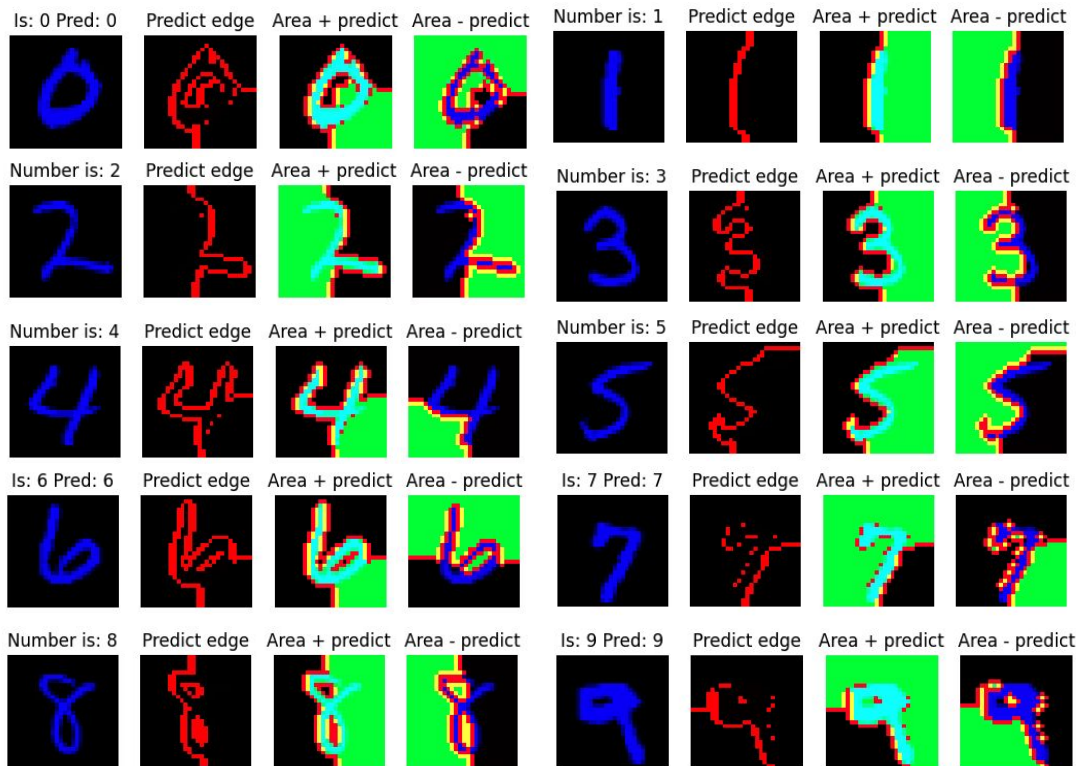


Dataset/dimension	Method	Accuracy
MNIST/784 features (28*28) 60k training 10k testing 4 epoch training	CNN Model 0 (LeNet)	98.22%
	CNN Model 1 (w/ dropout)	98.02% (p=0.2)
	CNN Model 2,3 (tuned filter size)	98.02%/98.25%
	CNN Model 4 (add filter numbers)	98.34%

- Similar structure with LeNet
 - 2 conv layer (5x5, 10 and 20 filters)
 - 2 fully connected layer
- Training batch size: 64
- Training method: pytorch backpropagation with stochastic gradient descent
- Modifications for optimized performance on MNIST
 - Add dropout layer to prevent overfitting
 - Modify filter size to be more adaptive to MNIST image size
 - Add filter numbers in the network

CNN with LIME on MNIST

— Why CNN thinks x is x?



Blue: MNIST digits

Red: LIME explanation boundary

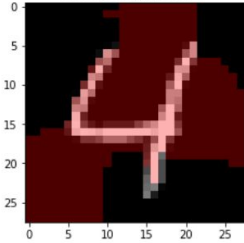

Green: indication area

Sky blue: overlap between digits and indication area

Yellow: overlap between MNIST digits and explanation boundary

- LIME has detected the digits that supports CNN prediction with high precision with the digits body.
- Some of the blank areas also included, while the support and against prediction area are complement to each other.

Results comparison

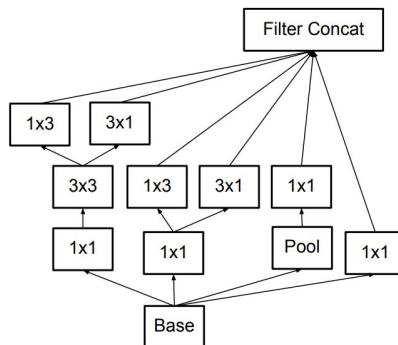
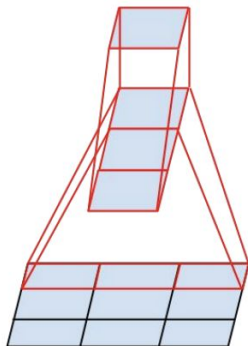
Methods	Random forest	LeNet based CNN mode
Accuracy	96.20%	98.34%
Data set	MNIST/784 features (28*28) (60k train 10k test)	
LIME results (colorful area)	<p>Positive Regions for 4</p> 	<p>Area + predict</p> 

Inception V3 with LIME on ImageNet

GoogLeNet(Inception Net/Inception V1): multi-branch heterogeneous architecture

Inception V2: 5x5 → two 3x3 convolution; Batch Normalization

Inception V3: $n \times n \rightarrow n \times 1$ and $1 \times n$ convolution (Factorized convolution)



"20": ["n01601694", "water_ouzel"],

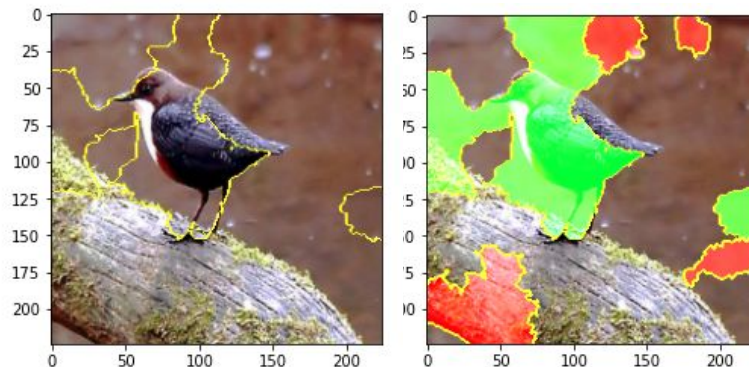
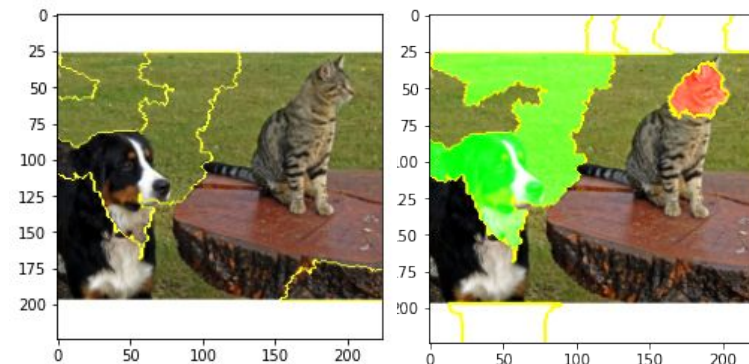


Image with dog and cat



Discussion and Summary



Conclusions

- We establish three machine learning models on 1D textual data (Comments) and 2D image data (MNIST) with high classification accuracy
- We evaluated LIME on the interpretation of the three machine learning models classification results

Problems

- LIME using simple model to approximate the local prediction property of complex model, limit itself to the linearity inversely
- Hard to evaluate the feature importance intuitively when the data dimension is high

In the next step...

- Use LIME to guide real world questions in medical fields
- Explore more methods to have better approximation...

References



- [1] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
- [2] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [3] Garreau, Damien, and Dina Mardaoui. "What does LIME really see in images?." *International Conference on Machine Learning*. PMLR, 2021.

Github repository Link

<https://github.com/chenchuw/CS542-FinalProject.git>





Thanks for listening and looking forward to questions!

Team 10 members

Bingquan Cai bqcai@bu.edu

Chuwei Chen chenchuw@bu.edu

Xiaowei Ge xwge@bu.edu



Backup slides below

TF-IDF

TFIDF works by proportionally increasing the number of times a word appears in the document but is counterbalanced by the number of documents in which it is present.

```
documents = [  
    "apple orange pear",  
    "apple apple pear",  
    "apple apple"  
]
```



	apple	orange	pear
0	0.425441	0.720333	0.547832
1	0.840802	0.000000	0.541343
2	1.000000	0.000000	0.000000

SVM Comment classifier dataset

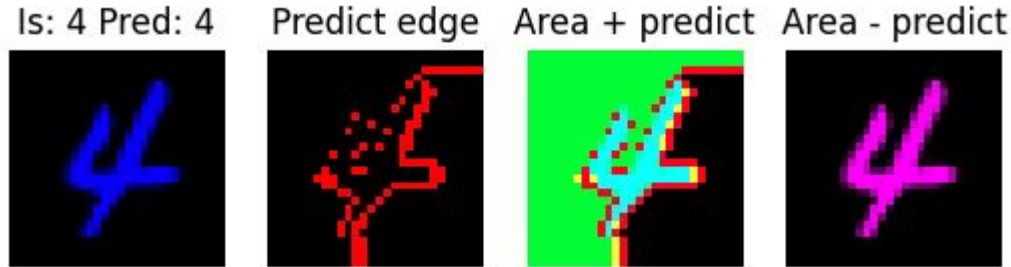
id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0000997932d777bf	Explanation	0	0	0	0	0	0
000103f0d9cfb60f	D'aww! He matches this t	0	0	0	0	0	0
000113f07ec002fd	Hey man, I'm really not tn	0	0	0	0	0	0
0001b41b1c6bb37e	"	0	0	0	0	0	0
0001d958c54c6e35	You, sir, are my hero. Any	0	0	0	0	0	0
00025465d4725e87	"	0	0	0	0	0	0
0002bcb3da6cb337	COCKSUCKER BEFORE YO! I	1	1	1	0	1	0
00031b1e95af7921	Your vandalism to the Ma	0	0	0	0	0	0
00037261f536c51d	Sorry if the word 'nonsens	0	0	0	0	0	0
00040093b2687caa	alignment on this subject	0	0	0	0	0	0
0005300084f90edc	"	0	0	0	0	0	0
00054a5e18b50dd4	bbq	0	0	0	0	0	0
0005c987bdfc9d4b	Hey... what is it..	1	0	0	0	0	0
0006f16e4e9f292e	Before you start	0	0	0	0	0	0
00070ef96486d6f9	Oh, and the girl above sta	0	0	0	0	0	0
00078f8ce7eb276d	"	0	0	0	0	0	0
0007e25b2121310b	Bye!	1	0	0	0	0	0
000897889268bc93	REDIRECT Talk-Voydan Poj	0	0	0	0	0	0
0009801bd85e5806	The Mitsurugi point made	0	0	0	0	0	0
0009eaea3325de8c	Don't mean to bother you	0	0	0	0	0	0
000b08c464718505	"	0	0	0	0	0	0
000bfd0867774845	"	0	0	0	0	0	0
000c0df995809fa	"	0	0	0	0	0	0
000c6a3f0cd3ba8e	"	0	0	0	0	0	0
000cfce9f050d471	"	0	0	0	0	0	0
000eefc67a2c930f	Radial symmetry	0	0	0	0	0	0

```
# Extract toxic & non-toxic comments
# 143346 non-toxic comments and 16225 toxic comments in the dataset
# How toxic is defined: words that are toxic, obscene, threat, insult, identity hate
normal = df[(df['toxic'] == 0) & (df['severe_toxic'] == 0) & (df['obscene'] == 0) & (
    df['threat'] == 0) & (df['insult'] == 0) & (df['identity_hate'] == 0)]
offensive = df[(df['toxic'] == 1) | (df['severe_toxic'] == 1) | (df['obscene'] == 1) | (
    df['threat'] == 1) | (df['insult'] == 1) | (df['identity_hate'] == 1)]

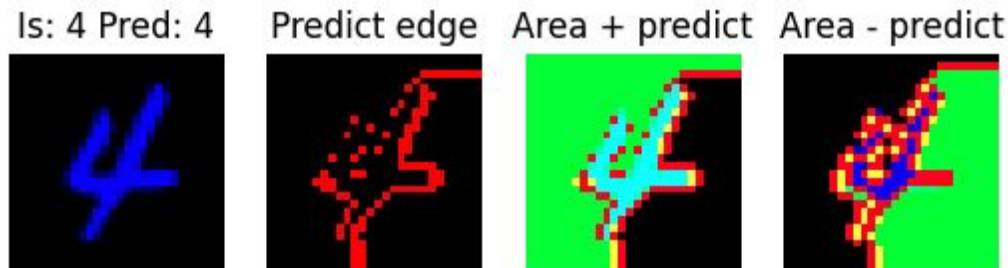
# Take 23891 non_toxic comments and 946 toxic comments
normal = normal[['id', 'comment_text', 'toxic']].iloc[:15000].copy()
offensive = offensive[['id', 'comment_text', 'toxic']].iloc[:15000].copy()

# Group toxic & non-toxic together
comments = pd.concat([normal, offensive], ignore_index=True)
```

Problem: LIME prediction on MNIST dataset

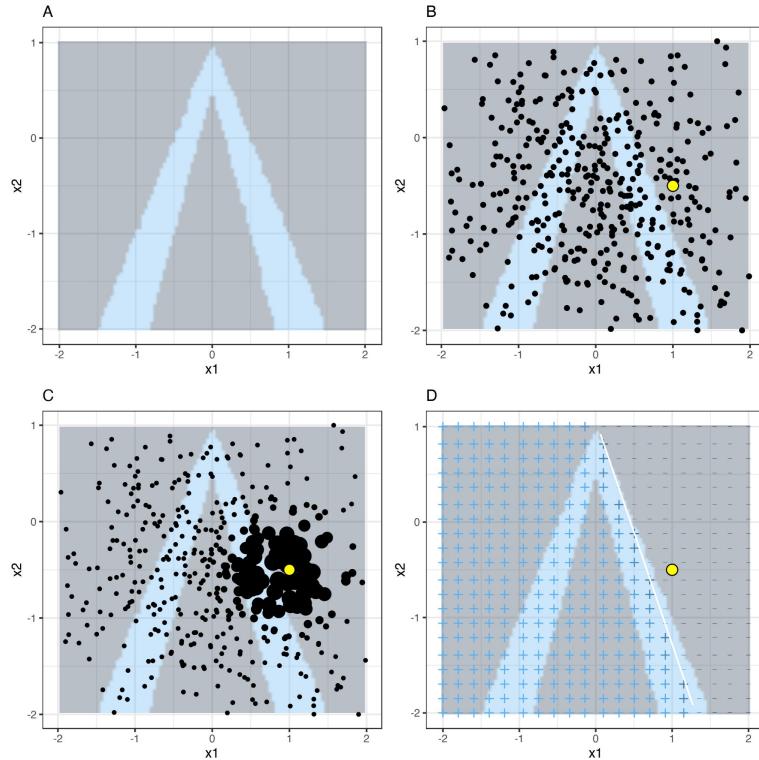


4 against
second top
possible
labels



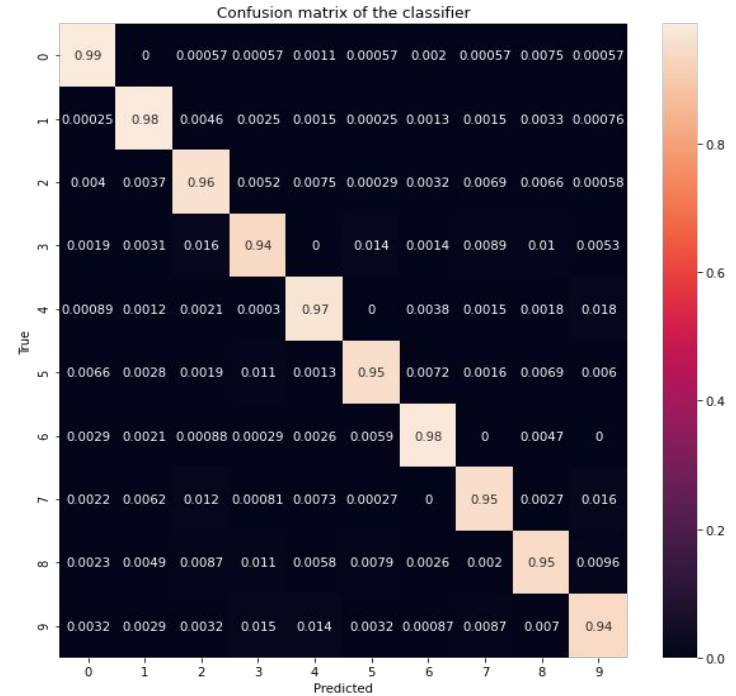
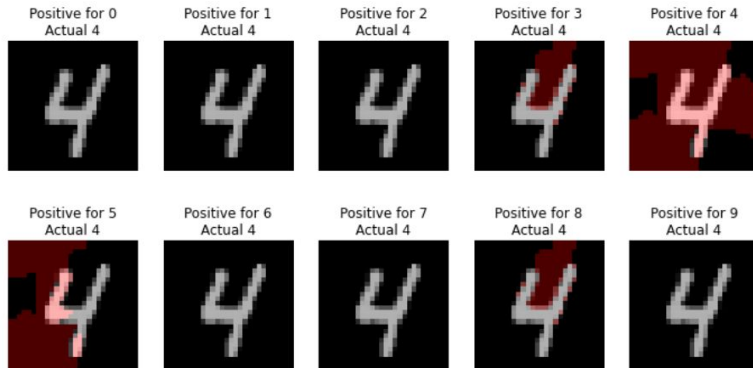
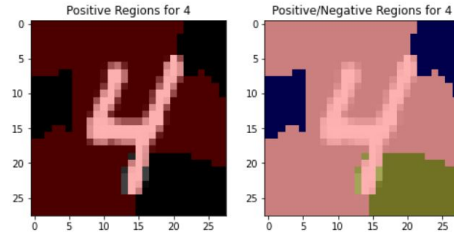
4 against rest
labels

Random forest with LIME



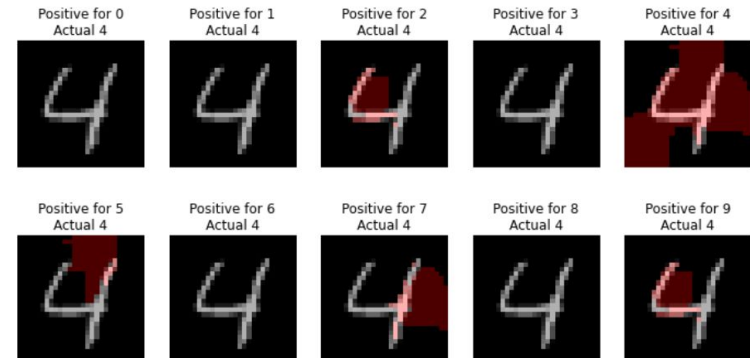
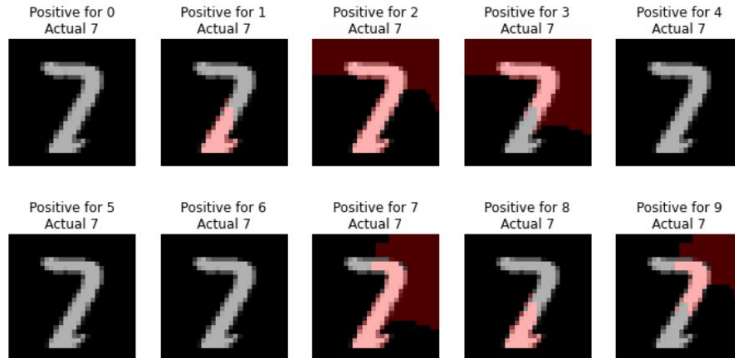
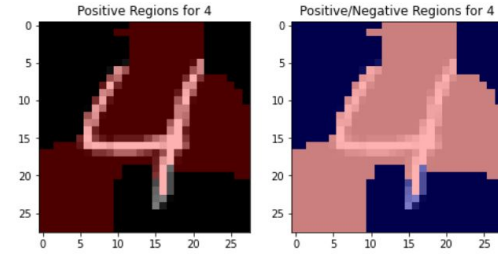
- Generated by perturbations
- Sampling from a normal distribution

Random forest with LIME

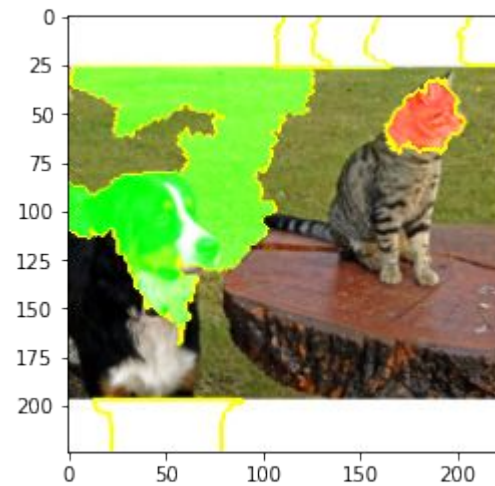
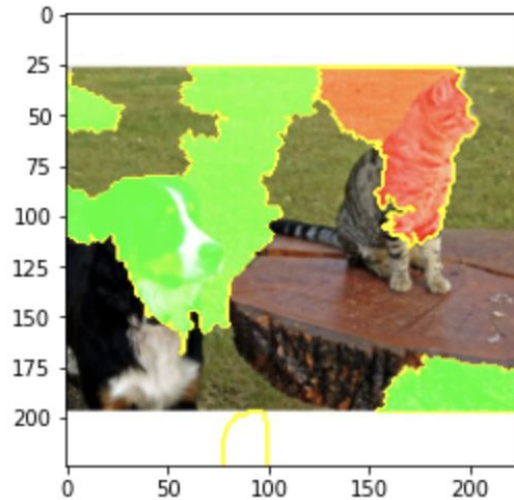


Random forest with LIME

Wrong prediction
 $7 \rightarrow 2$

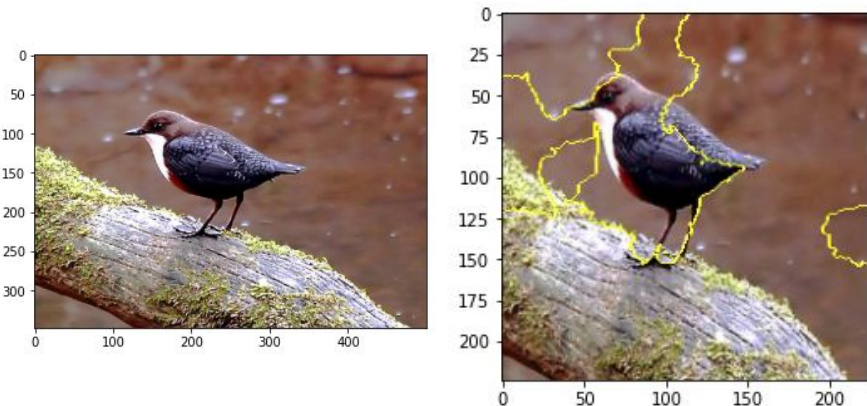


Randomness in LIME



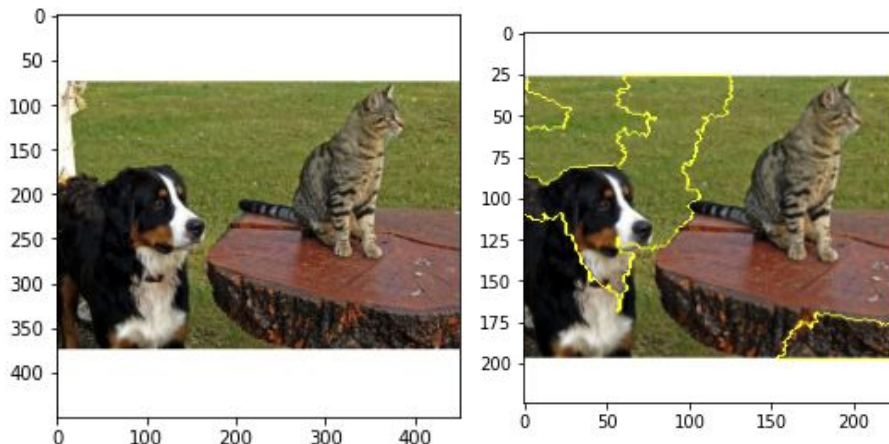
Inception V3 prediction results

"20": ["n01601694", "water_ouzel"],



```
((0.99999344, 20, 'water_ouzel'),
 (2.7273583e-07, 136, 'European_gallinule'),
 (2.0799166e-07, 995, 'earthstar'),
 (1.953845e-07, 86, 'partridge'),
 (1.9117357e-07, 98, 'red-breasted_merganser'))
```

Dog



```
((0.93592983, 239, 'Bernese_mountain_dog'),
 (0.038448066, 241, 'EntleBucher'),
 (0.023756348, 240, 'Appenzeller'),
 (0.0018181865, 238, 'Greater_Swiss_Mountain_dog'),
 (9.113341e-06, 214, 'Gordon_setter'))
```

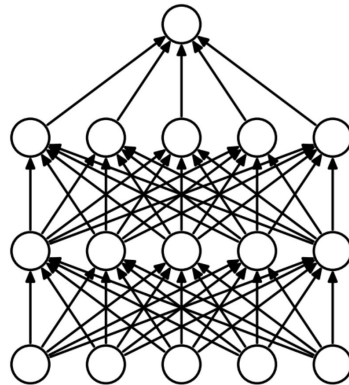
Inception roadmap



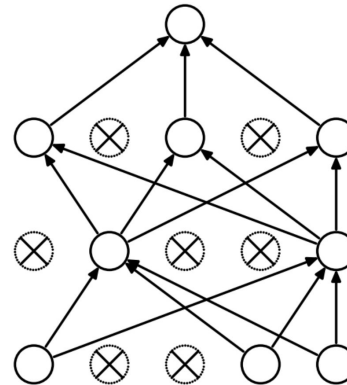
Network	Top-1 Error	Top-5 Error	Cost Bn Ops
GoogLeNet [20]	29%	9.2%	1.5
BN-GoogLeNet	26.8%	-	1.5
BN-Inception [7]	25.2%	7.8	2.0
Inception-v3-basic	23.4%	-	3.8
Inception-v3-rmsprop RMSProp	23.1%	6.3	3.8
Inception-v3-smooth Label Smoothing	22.8%	6.1	3.8
Inception-v3-fact Factorized 7×7	21.6%	5.8	4.8
Inception-v3 BN-auxiliary	21.2%	5.6%	4.8

Dropout layer

What does dropout do?



(a) Standard Neural Net

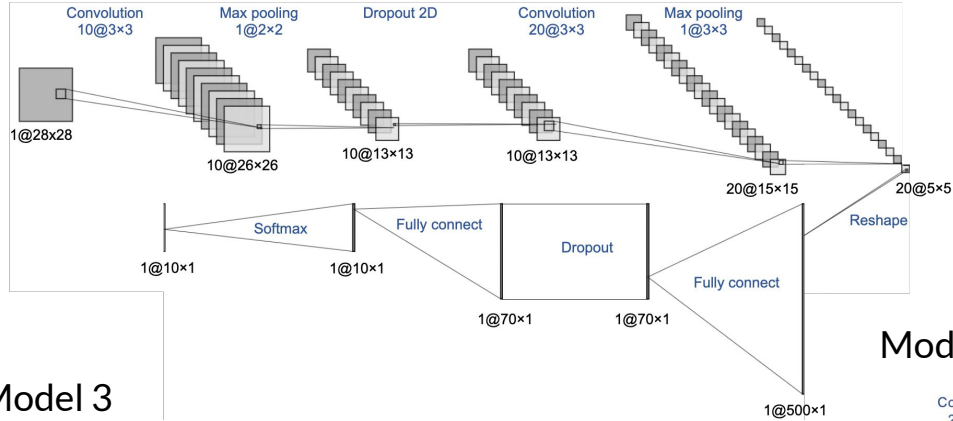


(b) After applying dropout.

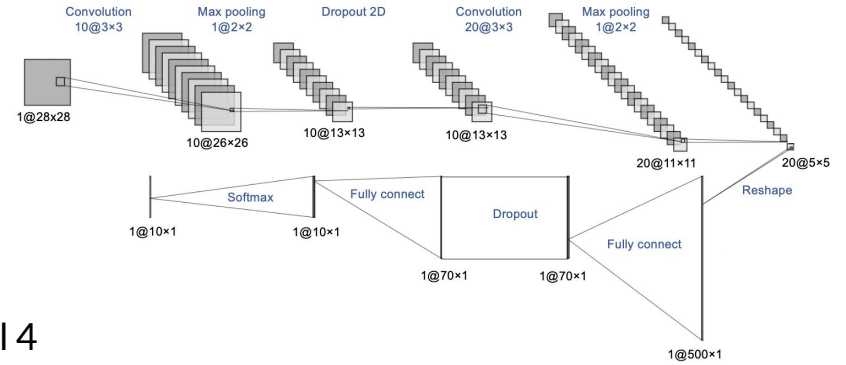
- Prevent overfitting
- Random subsampling

4 CNN models

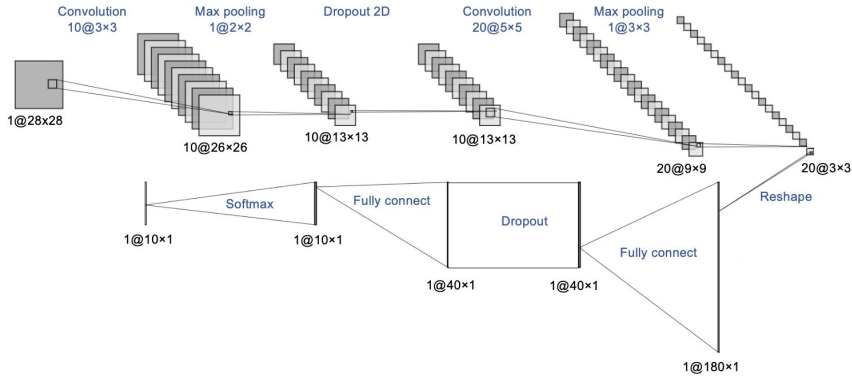
Model 1



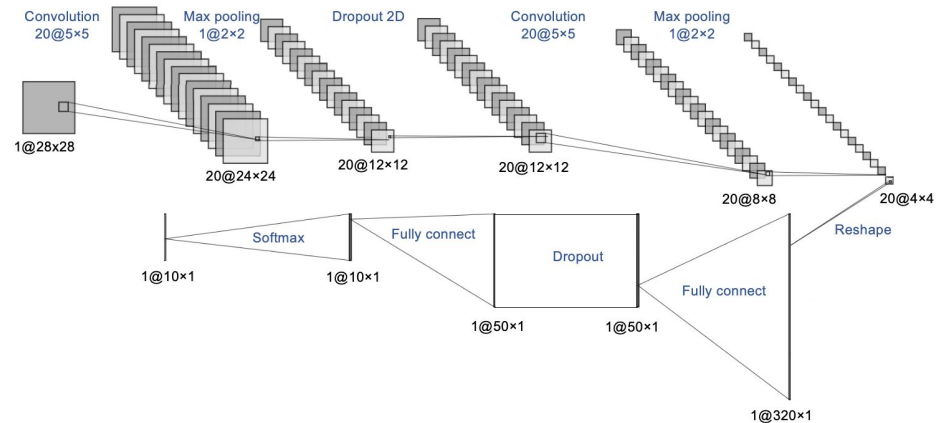
Model 2



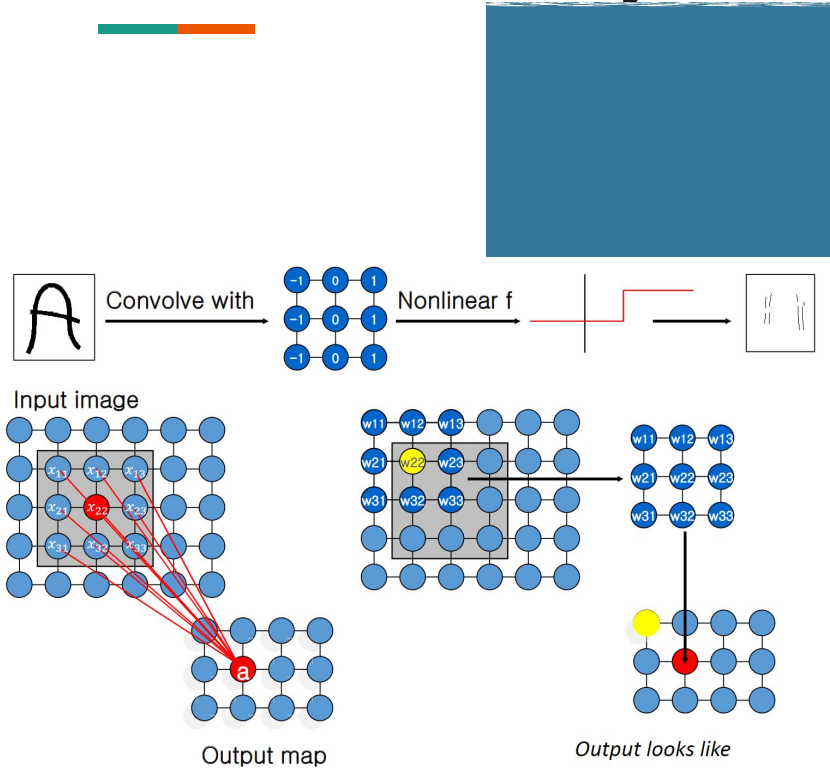
Model 3



Model 4

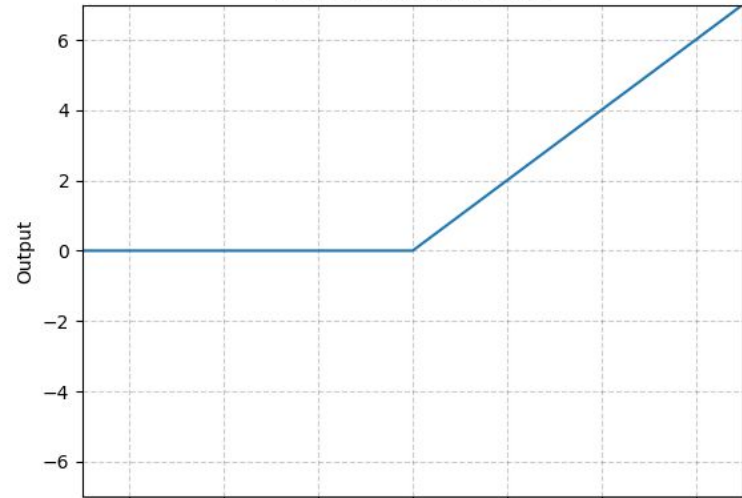


Convolutional Layer

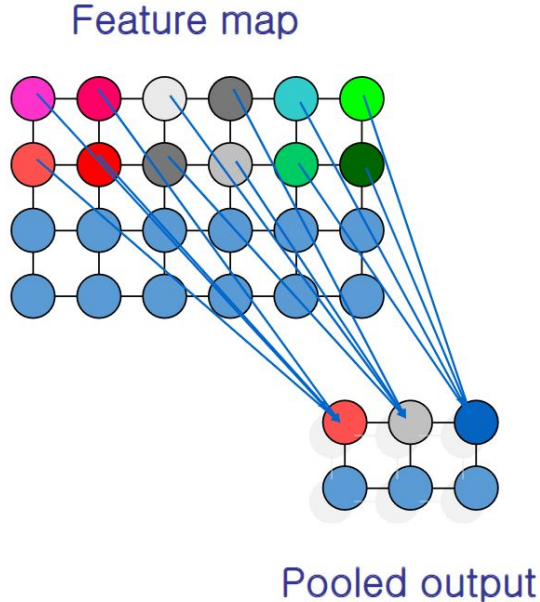


Relu

$$\max(0, x)$$



Pooling Layer



Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

σ = softmax

\vec{z} = input vector

e^{z_i} = standard exponential function for input vector

K = number of classes in the multi-class classifier

e^{z_j} = standard exponential function for output vector

e^{z_j} = standard exponential function for output vector

Input pixels, x



Shape: (3, 32, 32)

Feedforward output, y_i

	cat	dog	horse
5	5	4	2
4	4	2	8
4	4	4	1

Shape: (3,)

Softmax output, $S(y_i)$

	cat	dog	horse
	0.71	0.26	0.04
→	0.02	0.00	0.98
	0.49	0.49	0.02

Shape: (3,)

Forward
propagation

Softmax
function