Deep Active Learning using Monte Carlo Dropout

Lucas Moura
https://github.com/lucasmoura
lmoura@ime.usp.br

November 14, 2018

IME-USP: Institute of Mathematics and Statistics, University of São Paulo

Introduction

Motivation

- Deep Learning is a growing field with state-of-the-art results in several areas.
- Image Classification, Machine Translation

However...

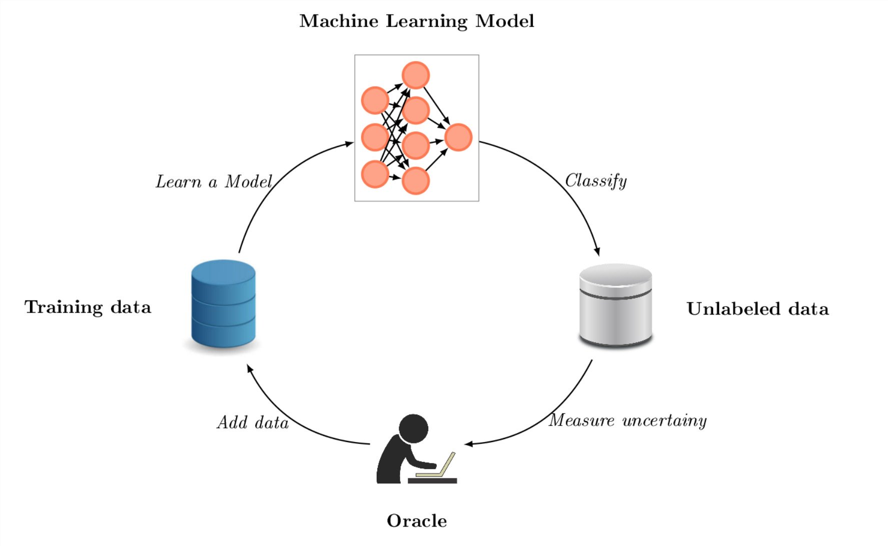
- Training Deep Learning models require a huge amount of labeled data
- For the task of image classification on the ImageNet database, 1.2 million labeled images were used [1]
- This restriction causes huge difficulties on applying Deep Learning techniques to a wide range of problems, such as Sentiment Analysis

Sentiment Analysis

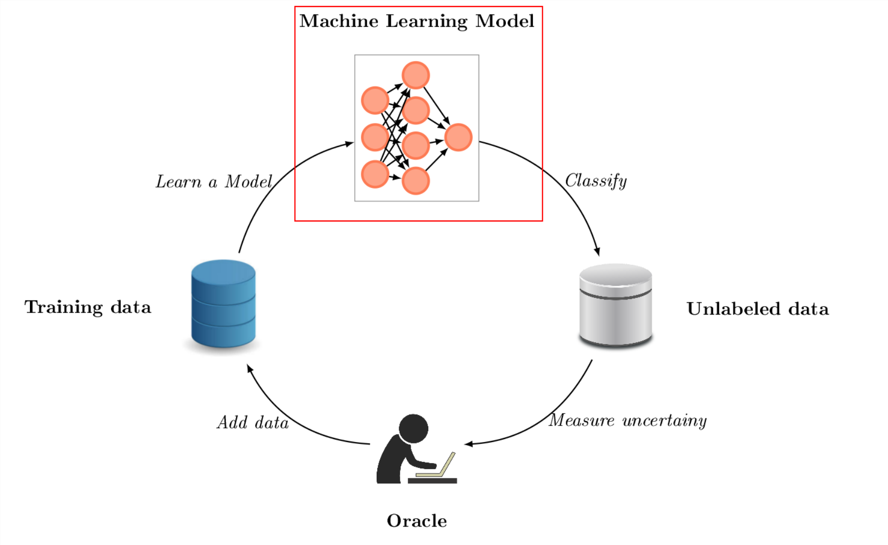
- Verify if a text is expressing negative or positive feelings.
- Huge amount of data, but few labeled.

Active Learning

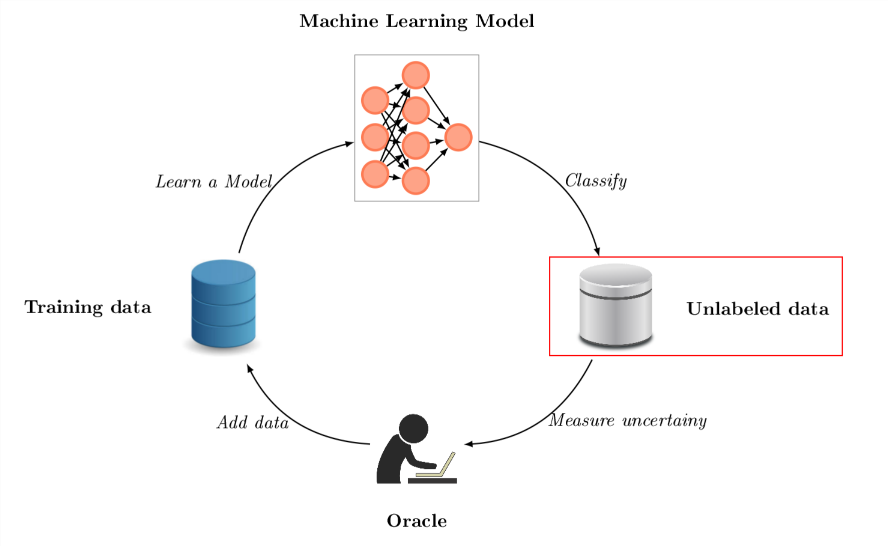
Active Learning



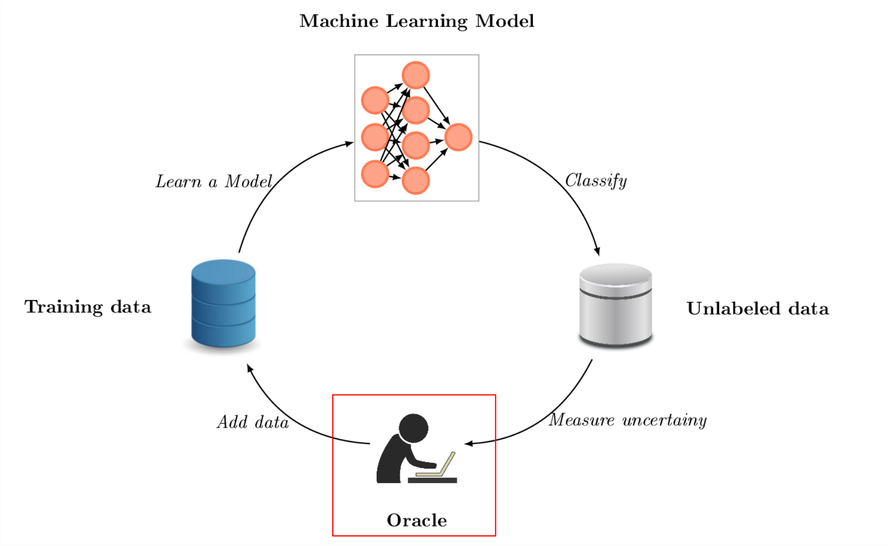
Active Learning



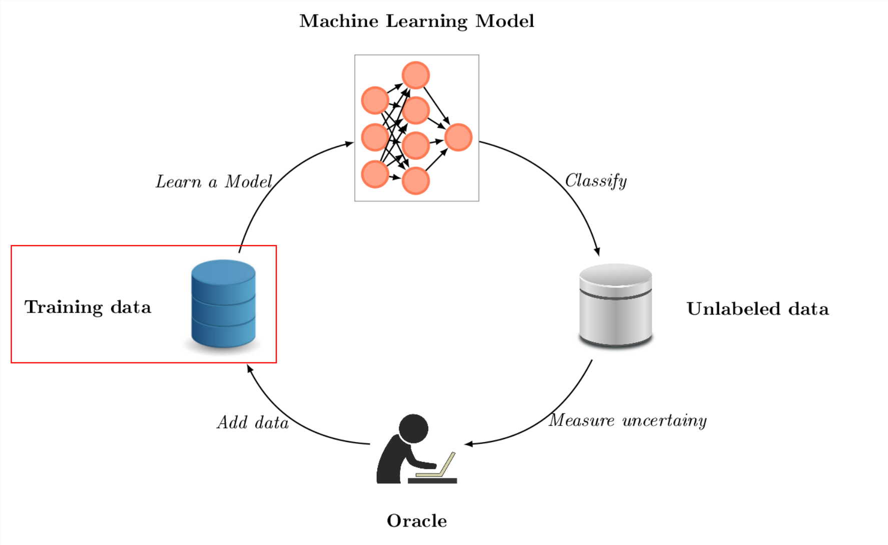
Active Learning



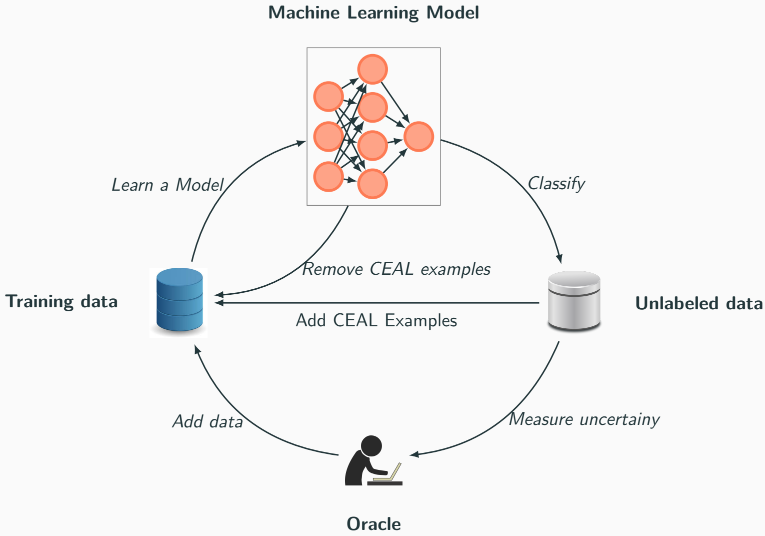
Active Learning



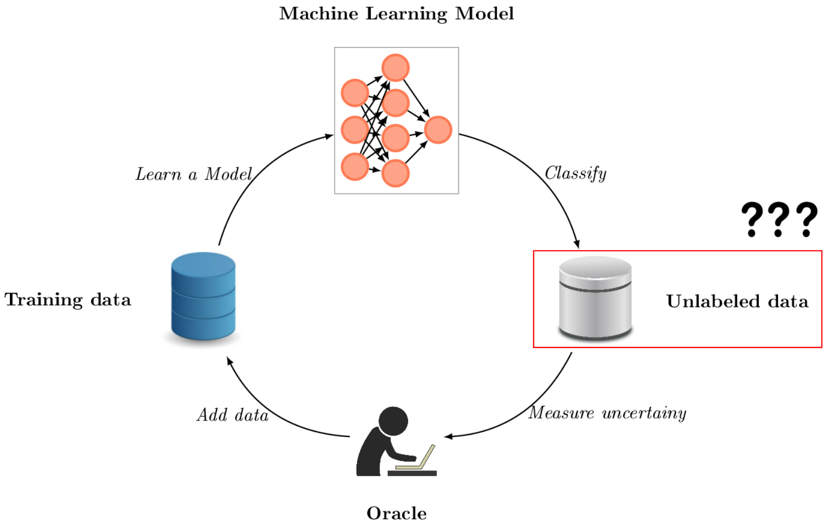
Active Learning



Active Learning - CEAL



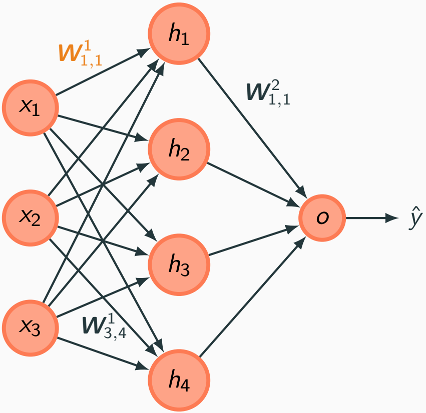
Active Learning



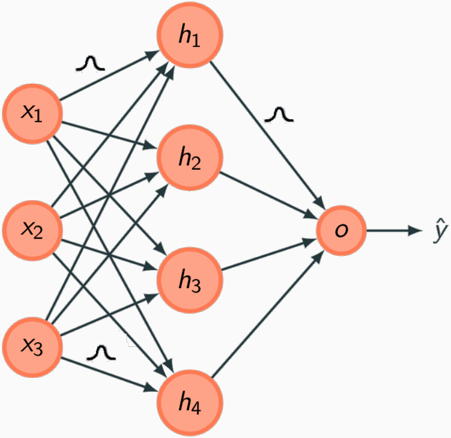
Uncertainty measurement

- To select informative samples, it is necessary to measure the uncertainty of the model prediction.

Neural Network



Bayesian Neural Network - prediction

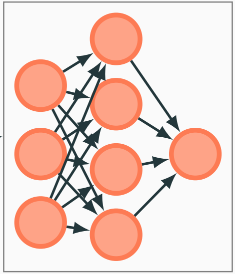


Bayesian Neural Network - prediction

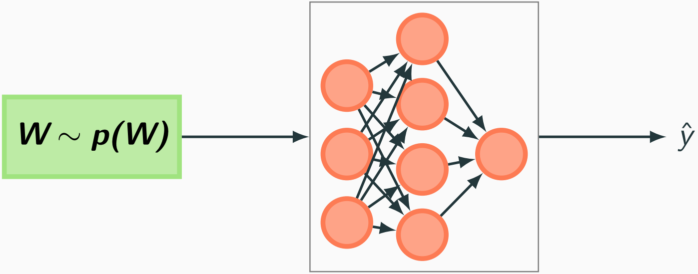
$$\boxed{W \sim p(W)}$$

Bayesian Neural Network - prediction

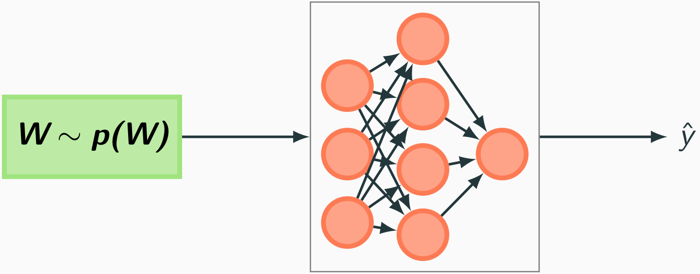
$W \sim p(W)$



Bayesian Neural Network - prediction

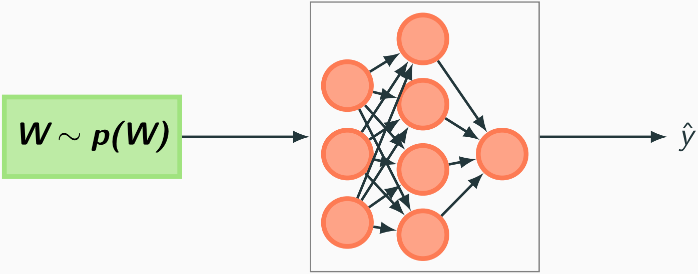


Bayesian Neural Network - prediction



Get T Classifications

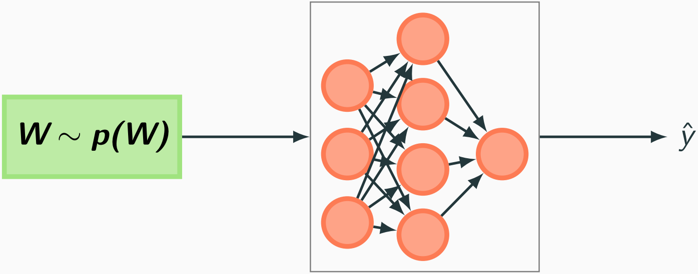
Bayesian Neural Network - prediction



Get T Classifications

$$Classifications = [\quad \hat{y}_1 \quad]$$

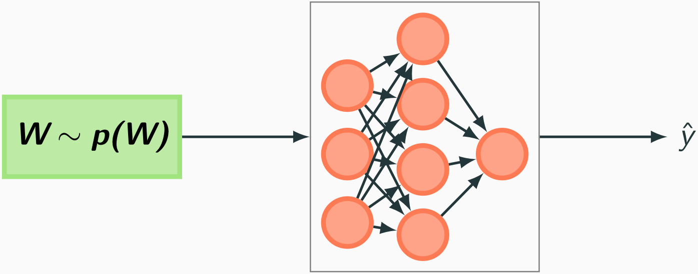
Bayesian Neural Network - prediction



Get T Classifications

$$\mathit{Classifications} = [\quad \hat{y}_1 \quad \hat{y}_2 \quad]$$

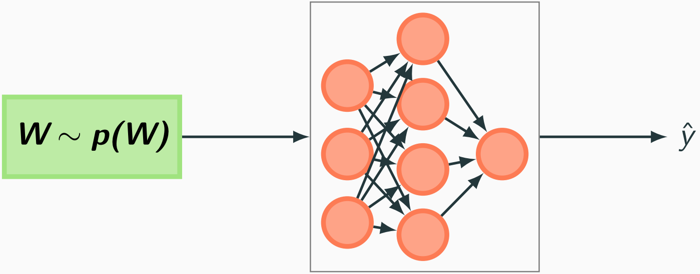
Bayesian Neural Network - prediction



Get T Classifications

$$Classifications = [\quad \hat{y}_1 \quad \hat{y}_2 \quad \hat{y}_3 \quad]$$

Bayesian Neural Network - prediction



Get T Classifications

$$Classifications = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \hat{y}_3 & \dots & \hat{y}_T \end{bmatrix}$$

Bayesian Neural Network - prediction

- Training Bayesian networks is a costly process
- Use techniques such as Variational Inference

Bayesian Neural Network

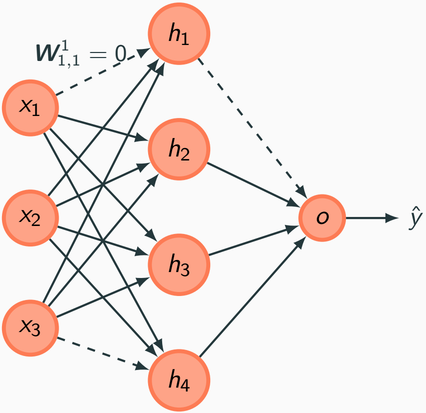
- What if we could extract uncertainty measurements from current Deep Learning models if they use stochastic regularization techniques such as Dropout ?
- Uncertainty in Deep Learning (Yarin Gal, 2017)

Dropout

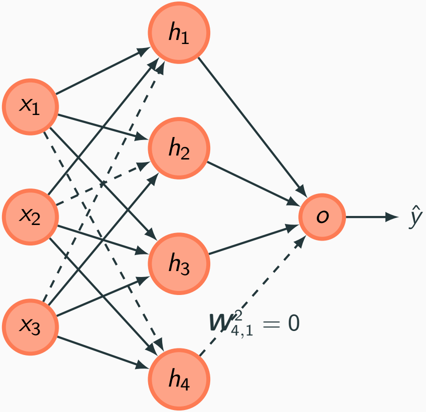
Dropout

- During training some weights are dropped from the network

Dropout



Dropout



Dropout

- The optimization function of Neural Networks using Dropout is practically the same as the optimization function of a Network trained with Variational Inference.
- Therefore it is possible to extract uncertainty measures from these networks, a technique called Monte Carlo Dropout.

Monte Carlo Dropout - Prediction

$\boxed{\boldsymbol{E \sim Bernoulli}}$

Monte Carlo Dropout - Prediction

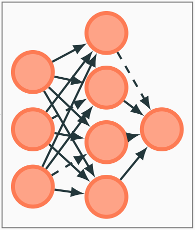
$$Dropout = \begin{bmatrix} 0 & 1 & \dots & 0 \end{bmatrix}$$

$\boldsymbol{E \sim Bernoulli}$

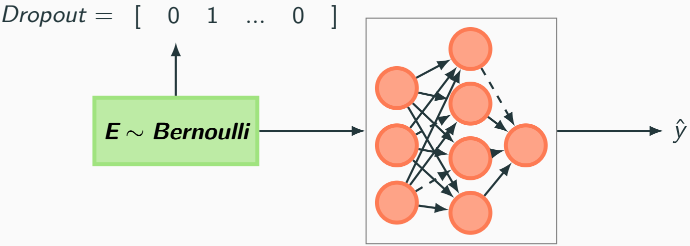
Monte Carlo Dropout - Prediction

$$Dropout = \begin{bmatrix} 0 & 1 & \ldots & 0 \end{bmatrix}$$

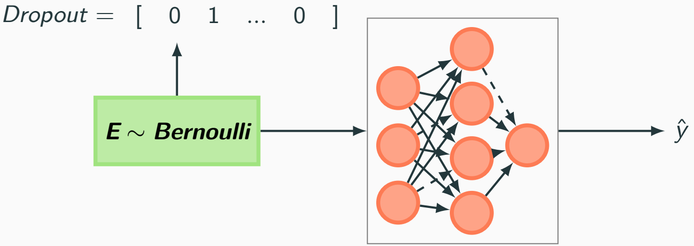
$\boldsymbol{E} \sim$ ***Bernoulli***



Monte Carlo Dropout - Prediction



Monte Carlo Dropout - Prediction

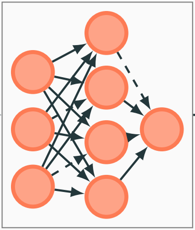


Get T Classifications

Monte Carlo Dropout - Prediction

$$Dropout = [\ 0 \quad 1 \quad \dots \quad 0\ ]$$

$\boldsymbol{E \sim Bernoulli}$



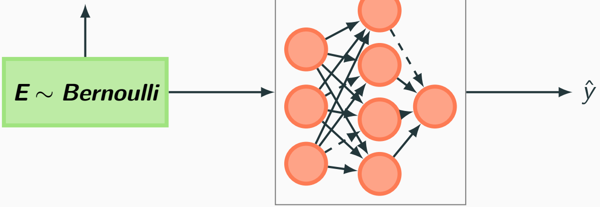
$\hat{y}$

Get T Classifications

$$Classifications = [\ \hat{y}_1\ ]$$

Monte Carlo Dropout - Prediction

$$Dropout = \begin{bmatrix} 0 & 1 & \ldots & 0 \end{bmatrix}$$

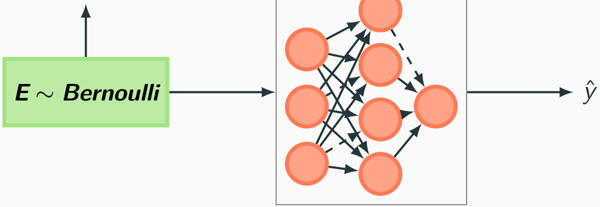


Get T Classifications

$$Classifications = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 \end{bmatrix}$$

Monte Carlo Dropout - Prediction

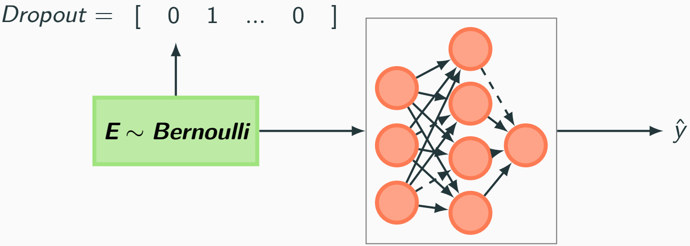
$$Dropout = \begin{bmatrix} 0 & 1 & \dots & 0 \end{bmatrix}$$



Get T Classifications

$$Classifications = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \hat{y}_3 \end{bmatrix}$$

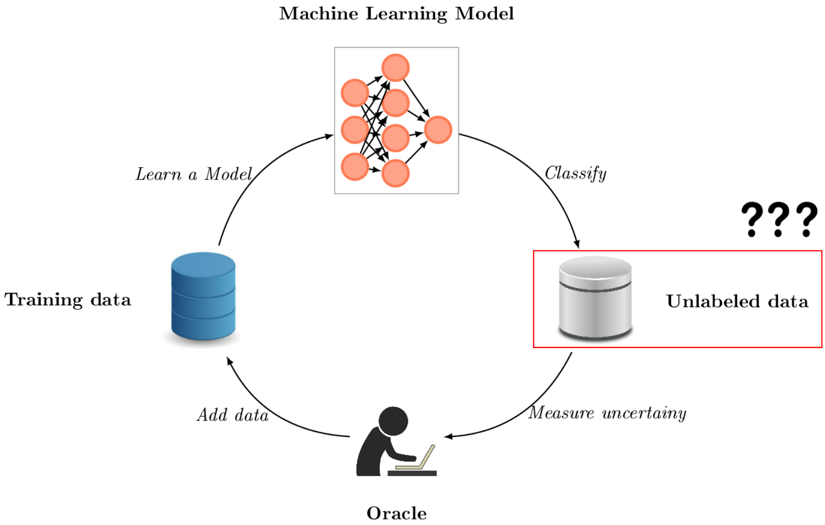
Monte Carlo Dropout - Prediction



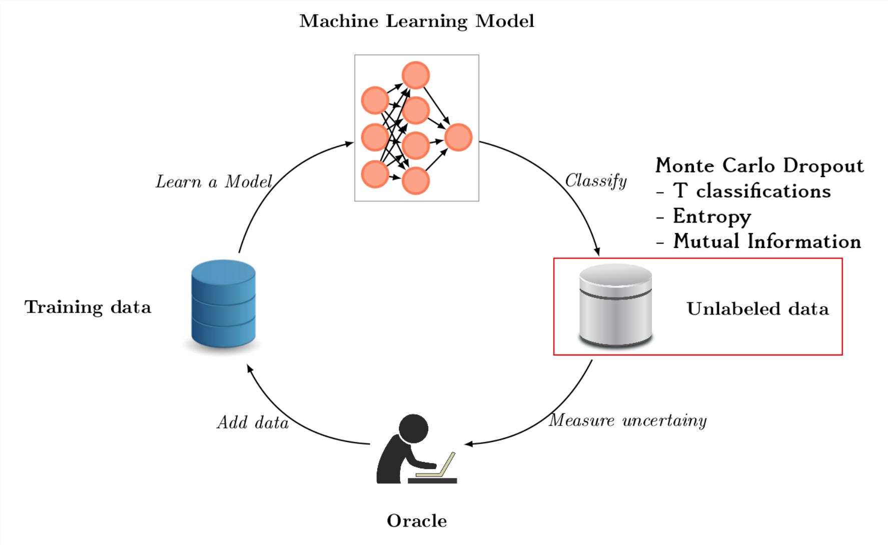
Get T Classifications

$$Classifications = [\ \hat{y}_1\ \ \hat{y}_2\ \ \hat{y}_3\ \ \ldots\ \ \hat{y}_T\ ]$$

Active Learning



Active Learning



Experimental Design

Research Contributions

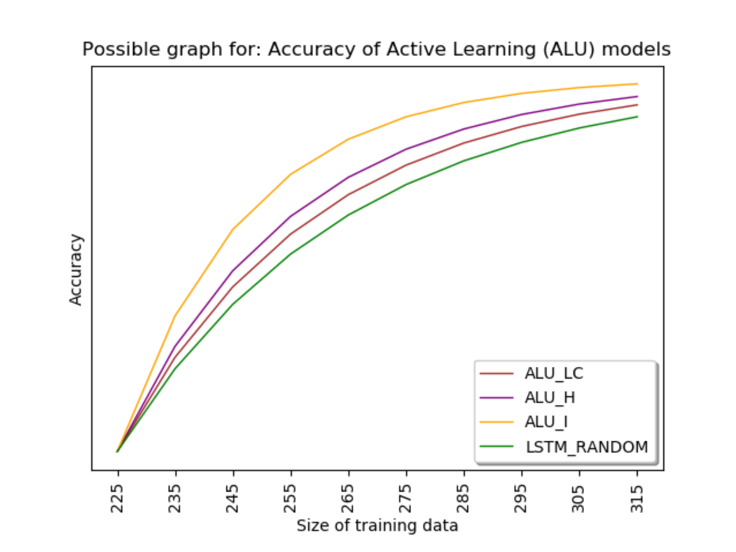
- An intrinsic comparison of Monte Carlo Dropout with random sampling strategies for the Active Learning framework for the task of sentiment analysis.
- An intrinsic comparison of Monte Carlo Dropout with the softmax uncertainty measurement for the Active Learning framework for the task of sentiment analysis
- Practical considerations when applying Deep Learning with Active Learning

Research Questions

- Q1: On the task of sentiment analysis, does modelling the uncertainty measurement of the model using the Monte Carlo Dropout technique help us achieve a better accuracy value in the Active Learning context ?

- Q2: Does Monte Carlo Dropout provides best uncertainty measurements then using the softmax output as a uncertainty measurement, the classical approach used in DL models ?

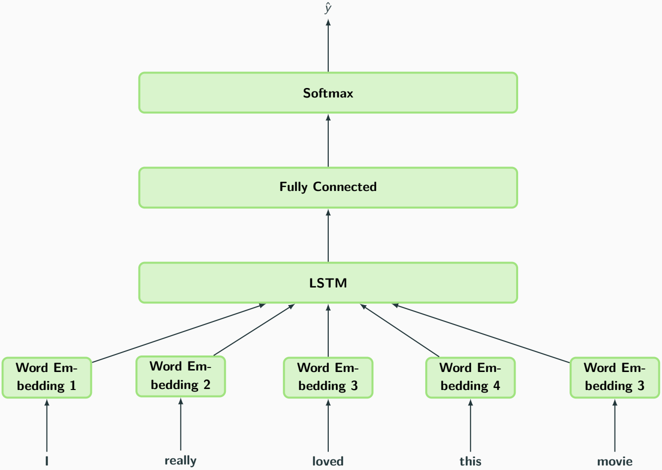
Active Learning



Datasets

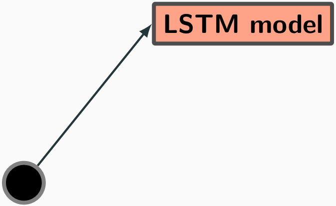
- Large Movie Review Dataset
  - 25000 train reviews and 25000 test reviews
  - Both train and test datasets have an equal number of positive and negative reviews
- Subjectivity Dataset
  - 10000 movie reviews divided into subjective and objective text.
  - Dataset perfectly balanced
  - Reviews hava an average size of 20 words

Network Archictecture

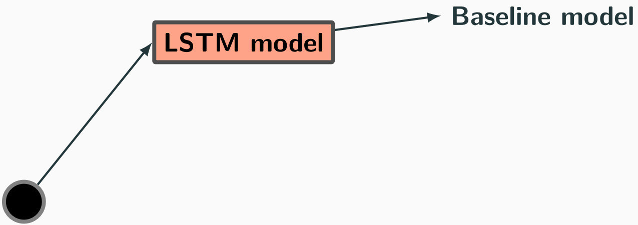


Experimental Design

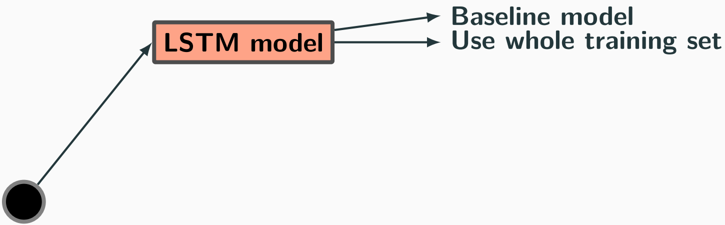
Experimental Design



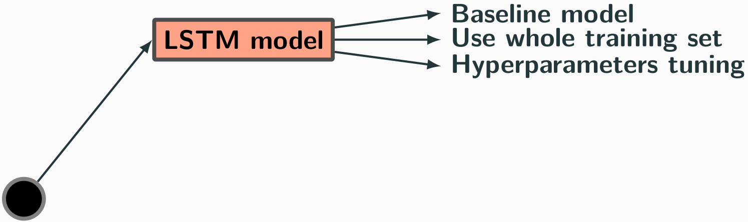
Experimental Design



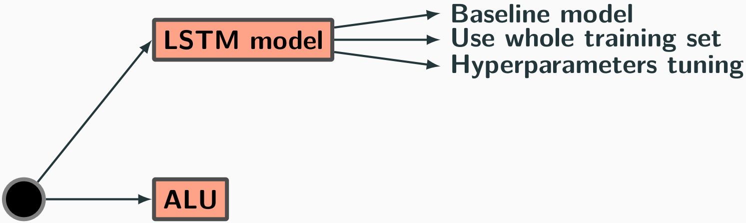
Experimental Design



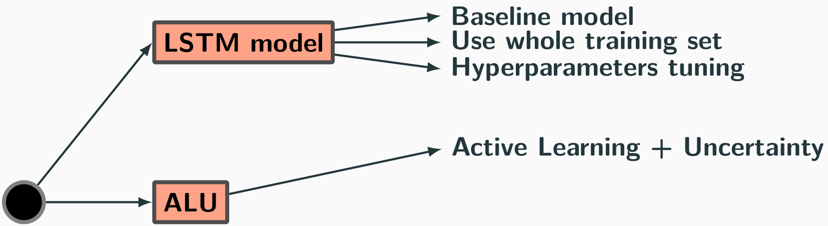
Experimental Design



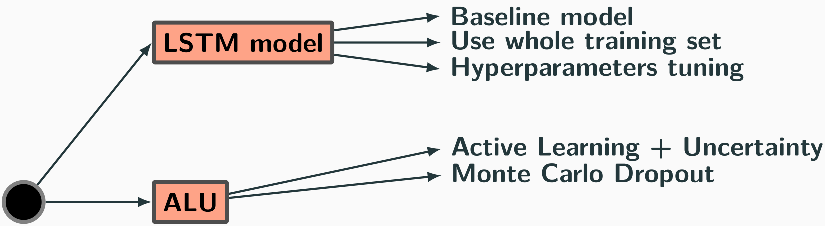
Experimental Design



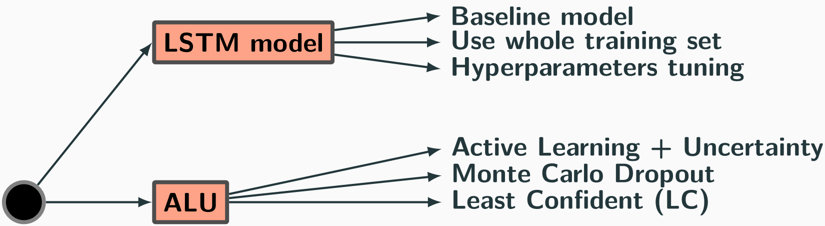
Experimental Design



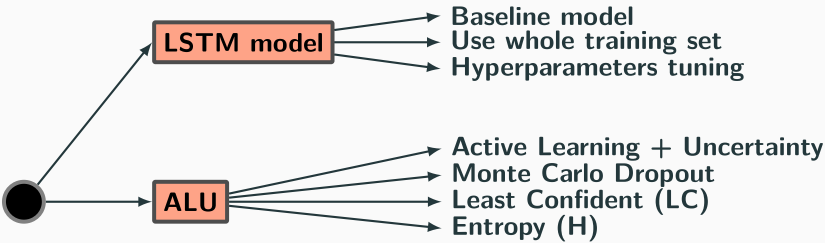
Experimental Design



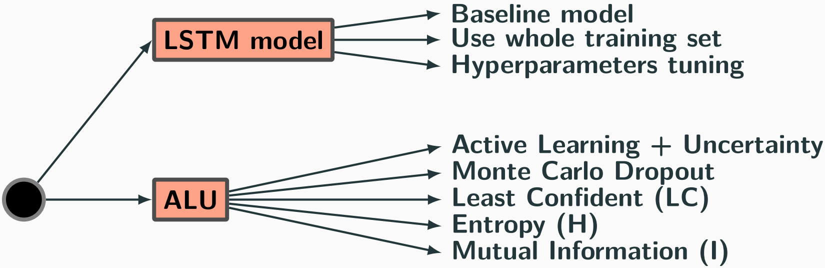
Experimental Design



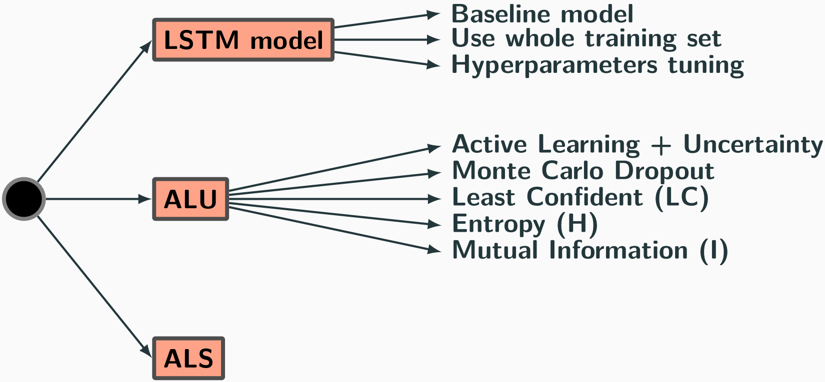
Experimental Design



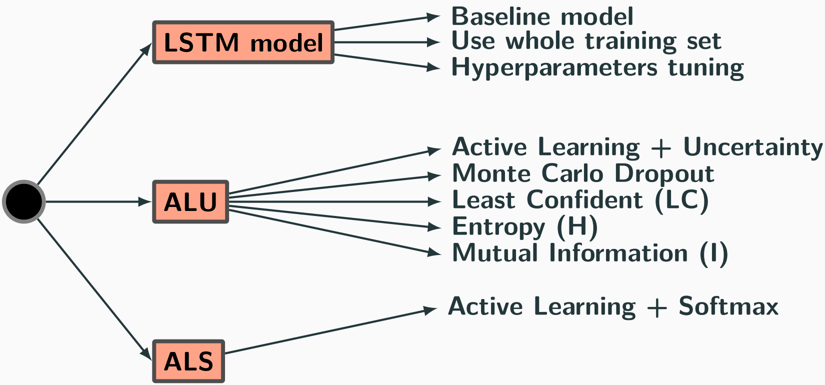
Experimental Design



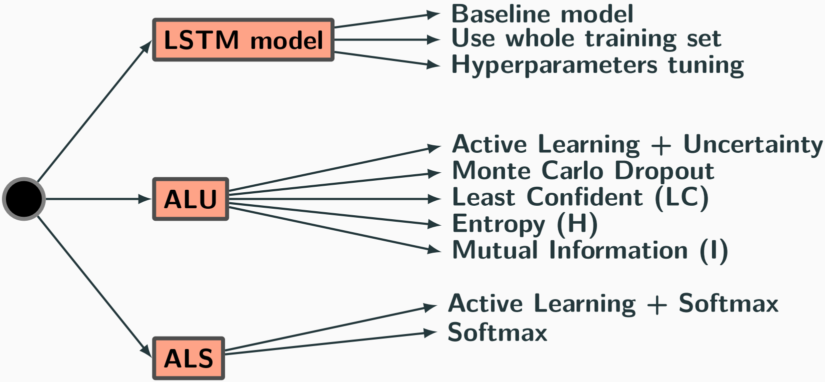
Experimental Design



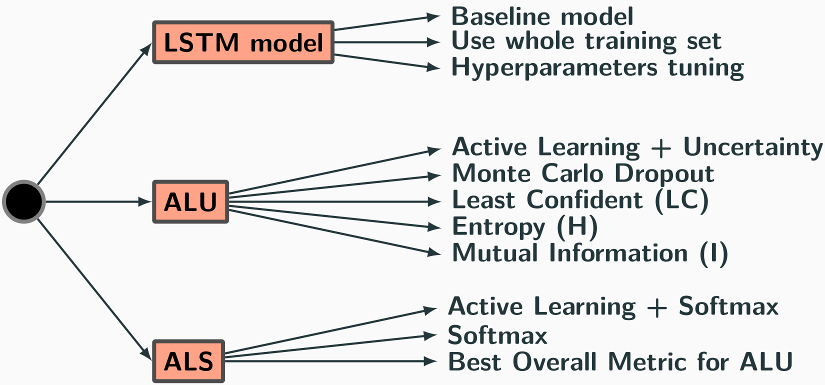
Experimental Design



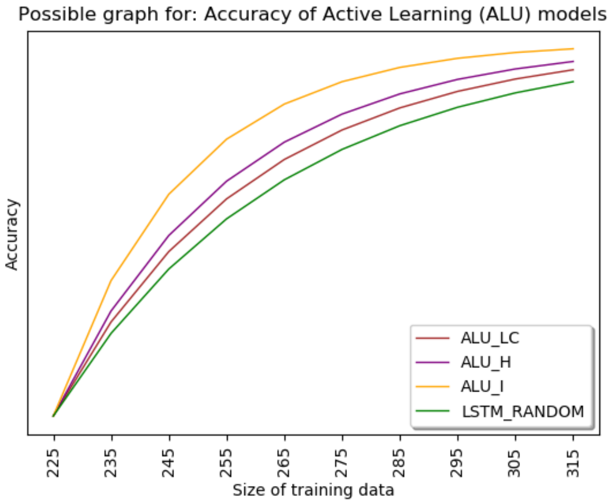
Experimental Design



Experimental Design



Active Learning



Active Learning Experiments Parameters

Dataset	Labeled Group	Unlabeled Group
Large Movie Review Dataset	225	22275
Subjectivity Dataset	10	8090

Active Learning Experiments Parameters

Parameter	Description
Unlabeled Data Queries (Q)	The number of example we will select from the unlabeled group to be labeled by the oracle.
Number of epochs (EPO)	At each AL cycle, we will train our model for a given number of epochs. This variable defines this quantity.
Dropout Values (DROP)	The dropout probability for the weights in our network.
Number of Active Learning Cycles (NC)	The number of AL cycles we have run for a given experiment.

Experiments Evaluation

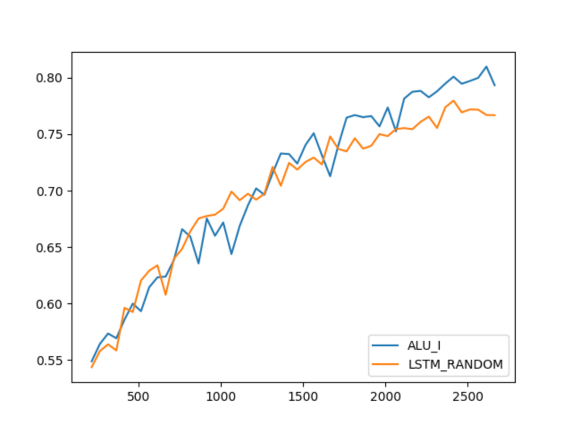
First Iteration

- Validate proposed model
- Find Hyperparameters
- Parameters
  - **Q (Number of data added to training dataset)** $= 50$
  - **EPO (Number of Epochs)** $= 16$
  - **NC (Number of Active Learning Cycles)** $= 50$
- Only one ALU model created (ALU_I)
- Random and ALU_I curve are almost identical

Second Iteration

- Increase number of epochs in experiments

Second Iteration

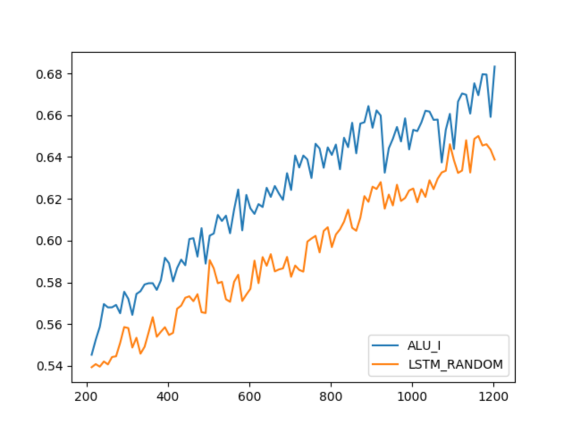


- **Q** = 50
- **EPO** = 200
- **NC** = 50

Third Iteration

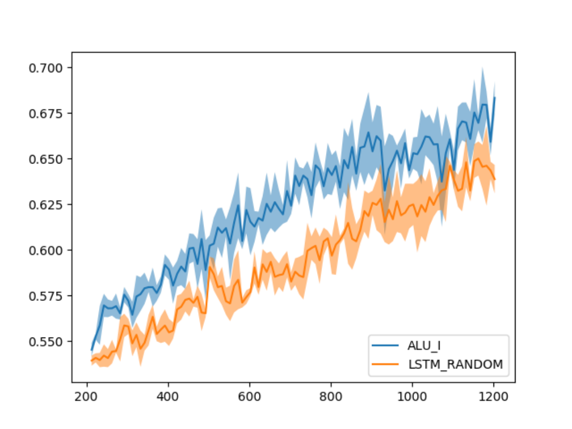
- Decrease number of unlabeled examples added to training dataset
- Decrease number of epochs
- Increase Dropour values

Third Iteration



- **Q** $= 10$
- **EPO** $= 150$
- **NC** $= 50$ **DROP** $= 0.5$

Third Iteration



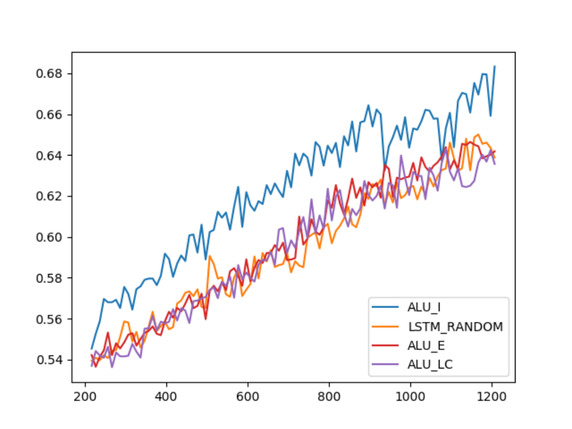
- **Q** = 10
- **EPO** = 150
- **NC** = 50

DROP = 0.5

Fourth Iteration

- Compare all metrics
- Use CEAL approach

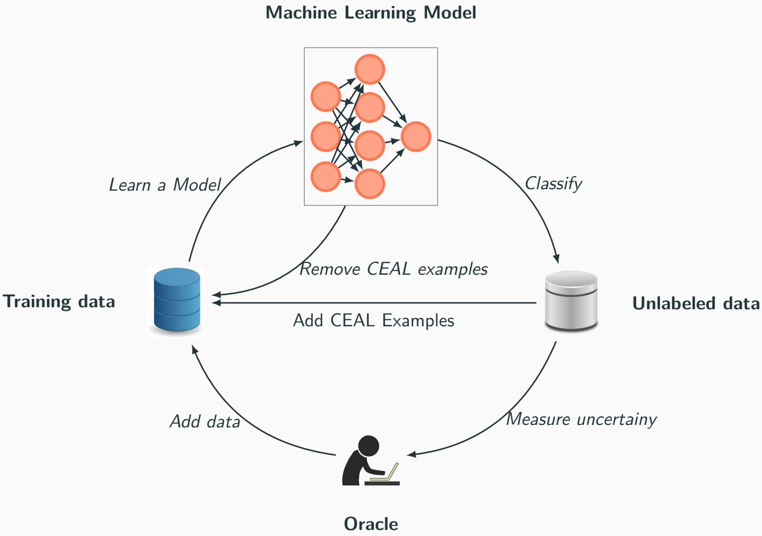
Fourth Iteration



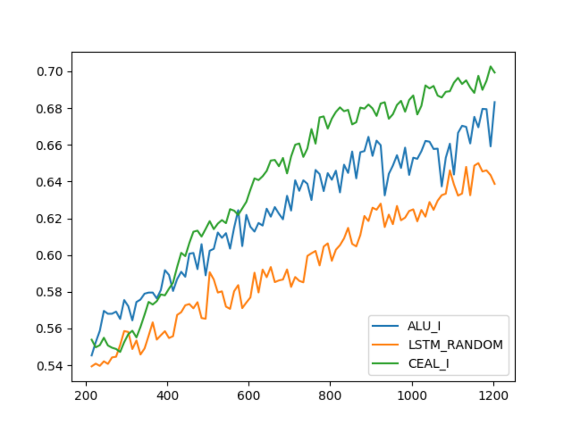
- **Q** $= 10$
- **EPO** $= 150$
- **NC** $= 100$

DROP $= 0.5$

Fourth Iteration



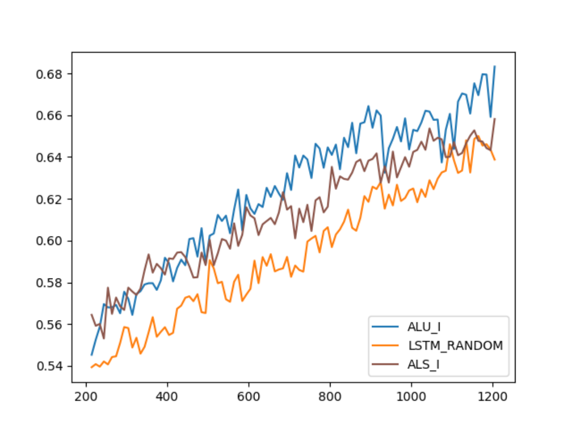
Fourth Iteration



Fifth Iteration

- Compare softmax metric

Fifth Iteration



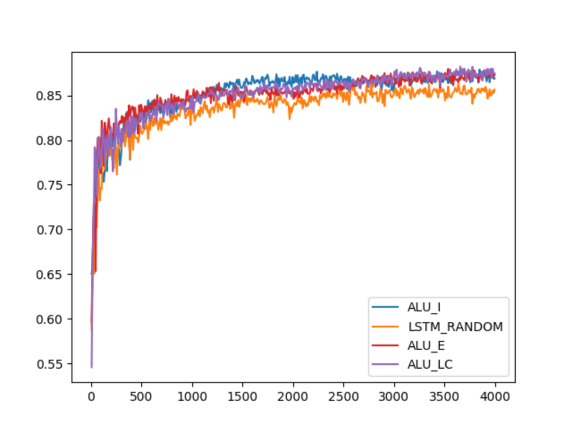
- **Q** = 10
- **EPO** = 150
- **NC** = 100

DROP = 0.5

Sixth Iteration

- Use Subjectivity Dataset
- Compare all metrics
- Use CEAL approach
- Make softmax comparison

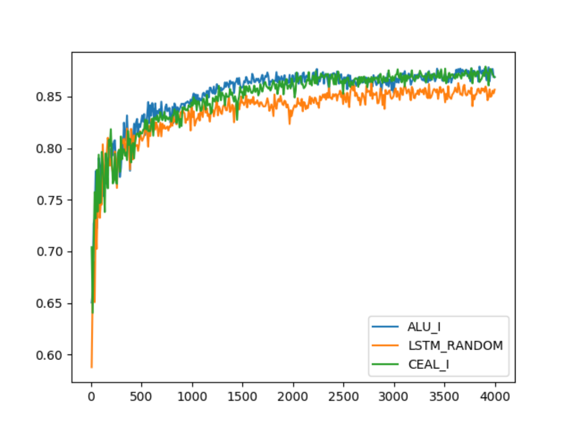
Sixth Iteration



- **Q** $= 10$
- **EPO** $= 150$
- **NC** $= 400$

DROP $= 0.5$

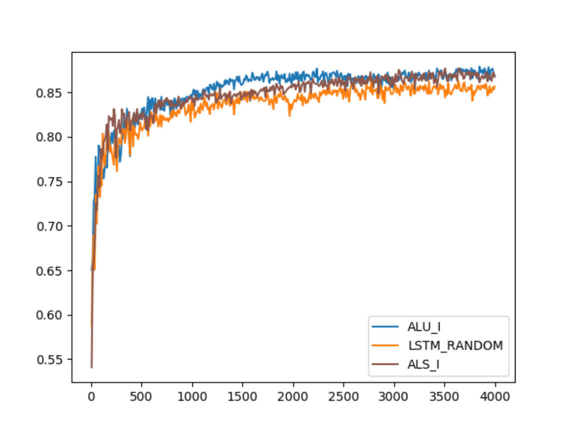
Sixth Iteration



- **Q** $= 10$
- **EPO** $= 150$
- **NC** $= 400$

DROP $= 0.5$

Sixth Iteration



- **Q** $= 10$
- **EPO** $= 150$
- **NC** $= 400$

DROP $= 0.5$

Conclusion

Conclusion

- Measuring the uncertainty of sample using the Monte Carlo Dropout has created better accuracy curves than using both random and softmax.
- We have positive results for both of our research questions
- Not consistent result for both datasets

However

Active Learning Problems

- LSTM is a bad archictecture for Active Learning
- Retraining Deep Learning algorithms each cycle does not scale
- Active Learning is biased approach
- Feel engineering approaches for Active Learning
- Costly framework and though to apply it to practical problems
- Huge number of variables to monitor

Future Work

- Explore CEAL framework with Monte Carlo Dropout for other areas, such as image recognition
- Evaluate more Active Learning parameters
- Monitor the types of sentences the model is choosing from the unlabeled group
- Engineering work

References

A. Krizhevsky, I. Sutskever, and G. E. Hinton.
Imagenet classification with deep convolutional neural networks.
Commun. ACM, 60(6):84–90, May 2017.