CSE/STAT 416 **Summer 2019**
Deep Learning Section 7

Concept Inventory

Here is a list of almost all the topics people wrote on their concept inventory. I have grouped them based on how I would categorize them, but this does not mean it is wrong to put them in different categories. Most topics appear in multiple places so this list would be long if we included every instance, but this is just how I think about them in my "ML Landscape". This is not meant to be an exhaustive list of topics, just big concepts that were listed on students' reflections.

Regression

- Linear Regression
- Polynomial Regression (linear regression with polynomial features)
- Quality Metric: RSS/MSE/RMSE
- Coefficients/weights/parameters
- Regularization
  - Ridge (L2 Penalty)
  - LASSO (L1 Penalty)
  - ElasticNet (L1 + L2)
- Local Methods
  - k-NN regression
  - Weighted k-NN regression
  - Kernel Regression
    - kernels (boxcar, Gaussian)
    - bandwidth

Classification

- Binary/multi-class classification
  - Sentiment analysis
  - Spam detection
  - Medical Diagnoses
- Majority class classifier
- Random Classifier
- Score Classifier
- Sigmoid Functions (Logistic function)
- Logistic regression
- Decision Trees
- Assessing Performance
  - Classification Error/Accuracy
  - True Positive, True Negative, False Positive, False Negative
  - Confusion matrix
  - TPR/FPR/etc.
  - Precision/Recall
- Decision Boundary
- Ensemble Methods
  - Random Forest (bagging)
  - AdaBoost or Gradient Boosting
- k-NN classifier

Document Retrieval

- k-NN Retrieval
- Clustering
- k-means/k-means++
- Hierarchical Clustering
  - Agglomerative
    - Single linkage
    - Complete linkage
  - Divisive
    - Recursive k-means
  - Dendrograms
- Similarity/Distance metrics
  - Euclidean Distance
  - Cosine Distance
  - Manhattan Distance
- Locality sensitive hashing
- Embeddings
  - Bag of words
  - TF-IDF

Misc

- ML Pipeline
- Supervised vs. unsupervised learning
- Feature vs input vs output
  - Polynomial features
- Model Complexity
- Overfitting/Underfitting
- Bias/Variance trade-off
- Hyperparameters
- Hyperparameter tuning
  - Grid search
  - Random search
- Train/Validation/Test
- Cross-Validation
  - k-fold validation
- Loss Function/Quality Metric
- Gradient Descent/Ascent
- Coordinate Descent/Ascent
- Maximum Likelihood Estimation (MLE)
- Feature engineering
- Scaling Features/Normalization
- One-hot encoding
- Missing data
- Curse of Dimensionality
- Greedy algorithms
- Efficiency ($O(n)$)
- Norms
  - L1 Norm
  - L2 Norm
- Dot Product/Inner Product

Convolution

1. Suppose we are given the following image:

1	2	1	3
4	3	3	3
5	5	4	4
1	1	2	4
3	2	1	5

Below, write out the result of doing a convolution with the following kernel. Assume we use no padding and a 1x1 stride.

0	1	2
0	0	2
0	0	1

Solution.

14	17
19	21
18	25

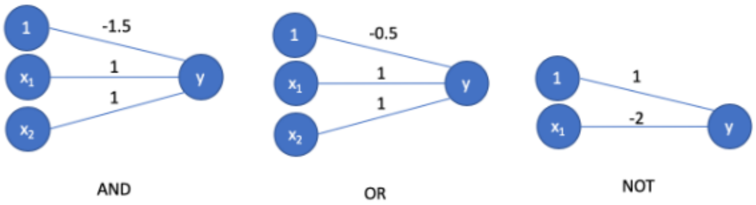
Neural Networks

1. Consider a neural network where each unit has a weight w and uses the hard threshold activation function defined as

$$h(x) = \begin{cases} 1 & , w^T x \geq 0 \\ 0 & , \text{ otherwise} \end{cases}$$

Construct the AND, OR gates with two inputs X_1 and X_2. Construct the NOT gate with X_1.

Solution.



2. (a) What is an epoch?

Solution. An epoch is when all of the data in the training set is presented to the neural network at once.

(b) What is a batch?

Solution. A batch is a subset of the data we get predictions for before back-propigating the differences through the network to update the weights.

(c) Given the algorithm defined in class

```
for i in range(num_epochs):
    for batch in batches(training_data):
        preds = model.predict(batch.data)
        diffs = compare(preds, batch.labels)
        model.backprop(diffs)
```

How many iterations of back-prop will we run? (answer as of a function of `num_epochs` and `num_batches`)

Solution. `num_epochs` $*$ `num_batches`