Natural Language Processing

Orevaoghene Ahia

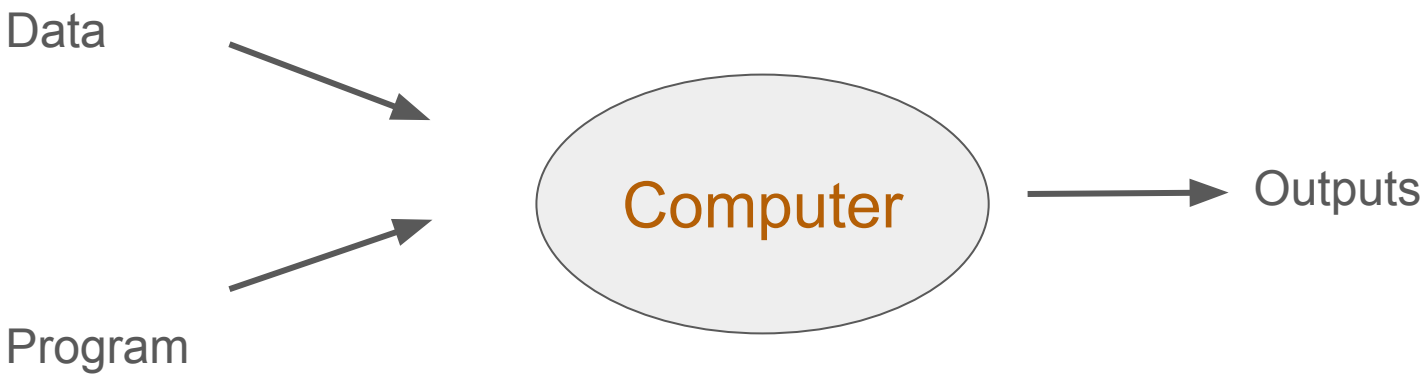
Lecture 5: REFRESHER ON CLASSIFICATION | TEXT CLASSIFICATION

What is Machine Learning ?

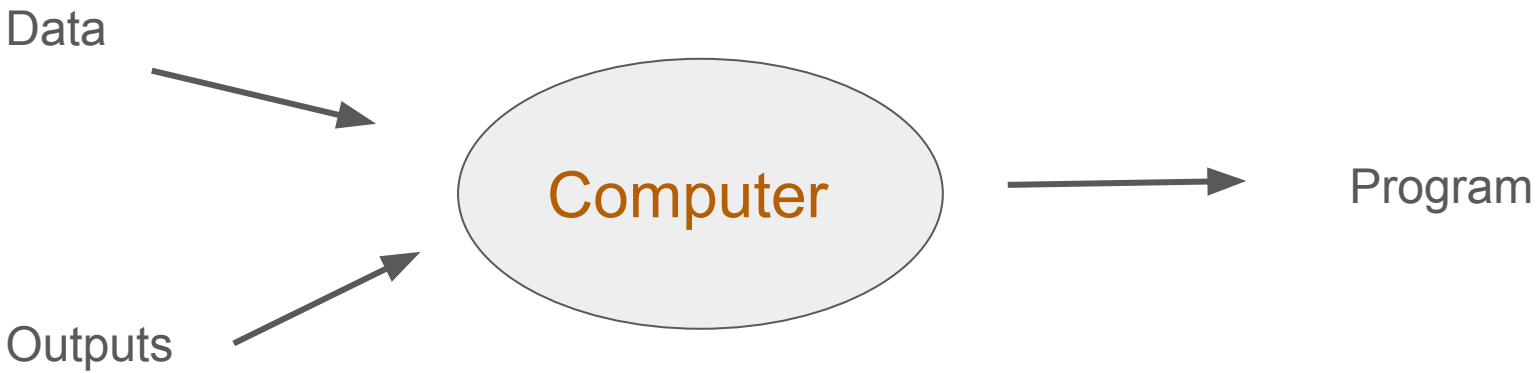
“Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed “

- Arthur L. Samuel, AI pioneer, 1959

Traditional Programming



Machine Learning



What are the categories of Machine Learning ?

Categories of Machine Learning

- **Supervised Learning**
  - **Given training data and desired outputs.**
  - **We feed the model with original outcomes, model learns and predicts outcome on new data**.
- Unsupervised Learning
  - Given training data without desired outputs.
  - The model makes sense of the data given to it.
- Reinforcement Learning
  - Reward for a sequence of actions .

Supervised Learning: Classification

- Classification is used to predict a **discrete class or label.**
- For example, predicting whether a person is likely to default on a loan or not is an example of a classification problem since the classes we want to predict are discrete: “likely to pay a loan” and “not likely to pay a loan”.

Supervised Learning: Regression

- Regression is used to predict a **continuous class or label.**
- A continuous output variable is a real-value, such as an integer or floating point value
- For example, where classification has been used to determine whether or not it will rain tomorrow, a regression algorithm will be used to predict the amount of rainfall.

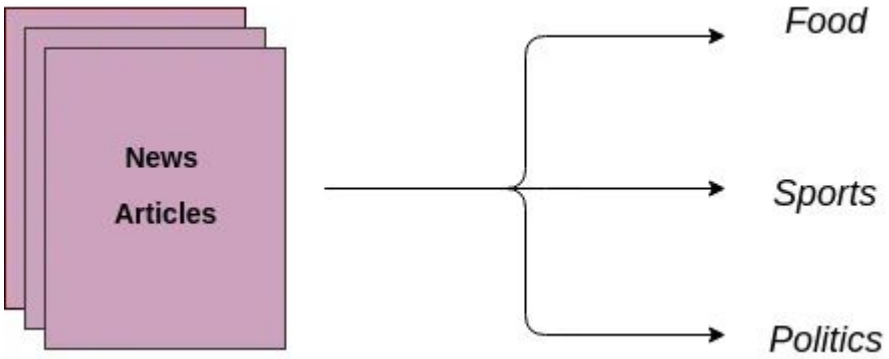
Give us 10 examples of classification and regression problems .

Regression vs Classification Problems

- Some supervised learning questions we can ask regarding movie data:
- Regression:
  - Can you predict the gross earnings for a movie ? $100 billion
- Classification:
  - Can you predict if a movie will win an oscar?
  - Can you predict the ratings of a movie.

Text Classification

- The process of assigning tags or categories in text according to its content.



Text classification can be grouped into

- Rule-based systems
- Machine Learning based systems
- Hybrid systems

- **Rule-based approaches** classify text into organized groups by using a set of handcrafted linguistic rules.
- These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content.

- For instance, to classify news articles into 2 groups; Sports and Politics.
  - You define a list of words that characterize each group.
  - Then you might classify the text, based on the counts of sports related words and politics related words. Classify as the group with the highest frequency .

- **Rule-based systems** are human comprehensible and can be improved over time.
- However, they require deep knowledge of the domain.
- They are also time-consuming, since generating rules for a complex system can be quite difficult and usually requires a lot of analysis and testing.
- Rule-based systems are also difficult to maintain and don’t scale well given that adding new rules can affect the results of the pre-existing rules.

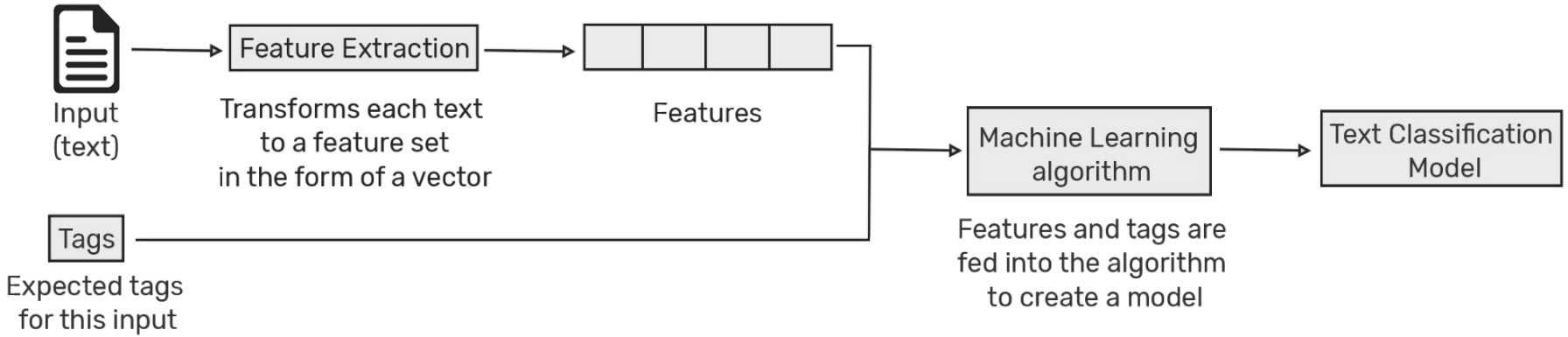
Machine Learning Based Systems

- Instead of relying on handcrafted rules, text classification with machine learning learns to make classifications based on previously observed data.
- A machine learning algorithm uses pre-labelled examples as training data to learn associations between pieces of text and that a particular output(tag) is for a particular input(Text).

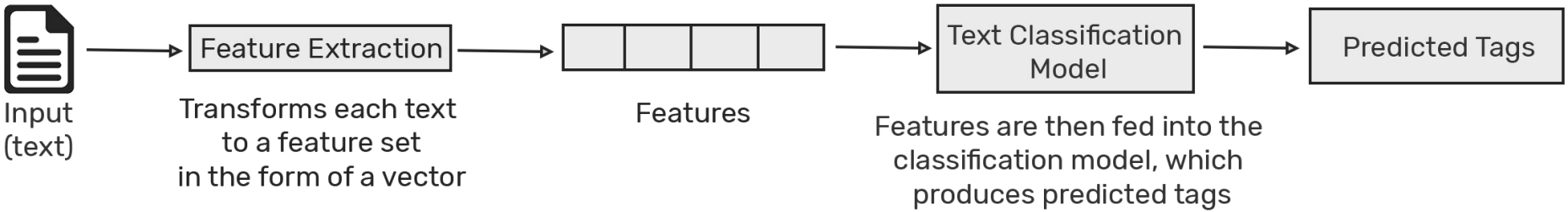
The first step towards training a classifier with machine learning is **feature extraction**: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words.

For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text *“This is awesome”*, we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0).

Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. *sports*, *politics*) to produce a classification model:



Once it's trained with enough training samples, the machine learning model can begin to make accurate predictions. The same feature extractor is used to transform unseen text to feature sets which can be fed into the classification model to get predictions on tags (e.g. *sports*, *politics*):



Text classification with machine learning is more accurate than handcrafted rule systems, it is also easier to maintain, as you can easily add tags to your new examples to learn on new tasks.

What are hybrid systems ?

Text Classification Algorithms

Logistic Regression

In NLP, logistic regression is the baseline supervised machine learning algorithm for classification, and also has a very close relationship with neural networks.

A neural network can be viewed as a series of logistic regression classifiers stacked on top of each other.

- A machine learning system for classification has 4 components:
  - A feature representation of the input . For each input observation x (i) the feature vector will be [x1, x2, x3...xn].
  - A classification function that computes ŷ, the estimated class, via p(y|x) (Sigmoid and softmax)
  - An objective function for learning, usually involving minimizing error on training examples.
  - An algorithm for optimizing the objective function. We use the Stochastic gradient descent algorithm.
- Logistic regression has 2 phases:
  - Training: We train the system using stochastic gradient and cross-entropy loss.
  - Test: Given a test examples x we compute p(y|x) and return the higher probability label y= 1 or y = 0.

The goal of binary logistic regression is to train a classifier that can make a binary decision about the class of a new input observation. The sigmoid classifier will help to make this decision.

- Consider a single input observation x, represented by a vector of features [x1, x2, x3...xn].
- The classifier output y can be 1 or 0
- We want to determine the probability P(y =1|x) that this observation is a member of that class.
- If the classes are positive sentiment and negative sentiment;
    - P(y=1|x) represents probability that the feature vector is a positive sentiment
    - P(y=0|x) represents probability that the feature vector is a negative sentiment

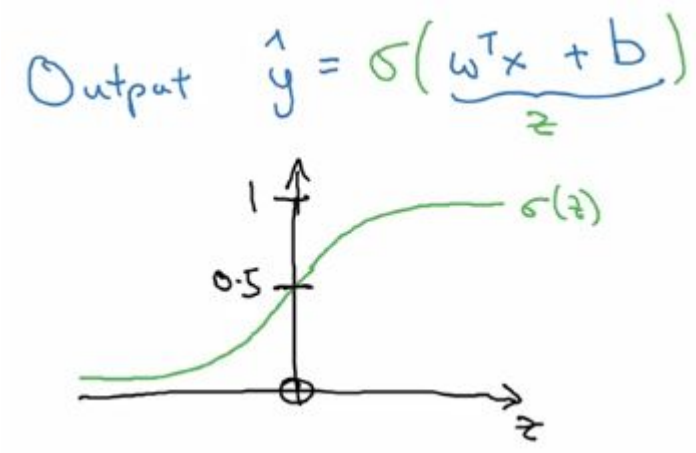
- Logistic regression solves this task by learning, from a training set, a vector of weights and a bias term.
- The weight w(i) represents how important that input feature is to the classification decision, and can be positive (meaning the feature is associated with the class) or negative (meaning the feature is not associated with the class)
- The bias term, also called the intercept intercept, is another real number that's added to the weighted inputs.

$W x + b = z$

For the output z to be a probability, we pass

it through a sigmoid function σ

$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$



After applying the sigmoid function on the sum of weighted features, the results is a number between 0 and 1

For a test instance x, we say yes if the probability P(y = 1|x) is more than .5, and no otherwise. We call .5 the decision boundary

$$\hat{Y} = \begin{cases} 1 \text{ if } p(y = 1 | x) > 0.5) \\ 0 \text{ if otherwise} \end{cases}$$

How are the parameters of the model, the weights w and bias b, learned?

We want to learn parameters (meaning w and b) that make ŷ for each training observation as close as possible to the true y .This requires two components:

Loss function: Cross-entropy (conditional maximum likelihood estimation)

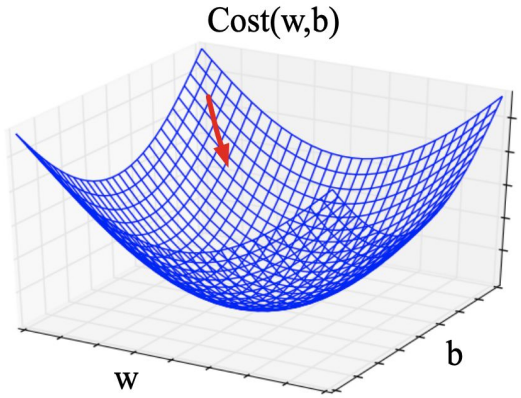
(we choose the parameters w,b that maximize the log probability of the true y labels in the training data given the observations x)

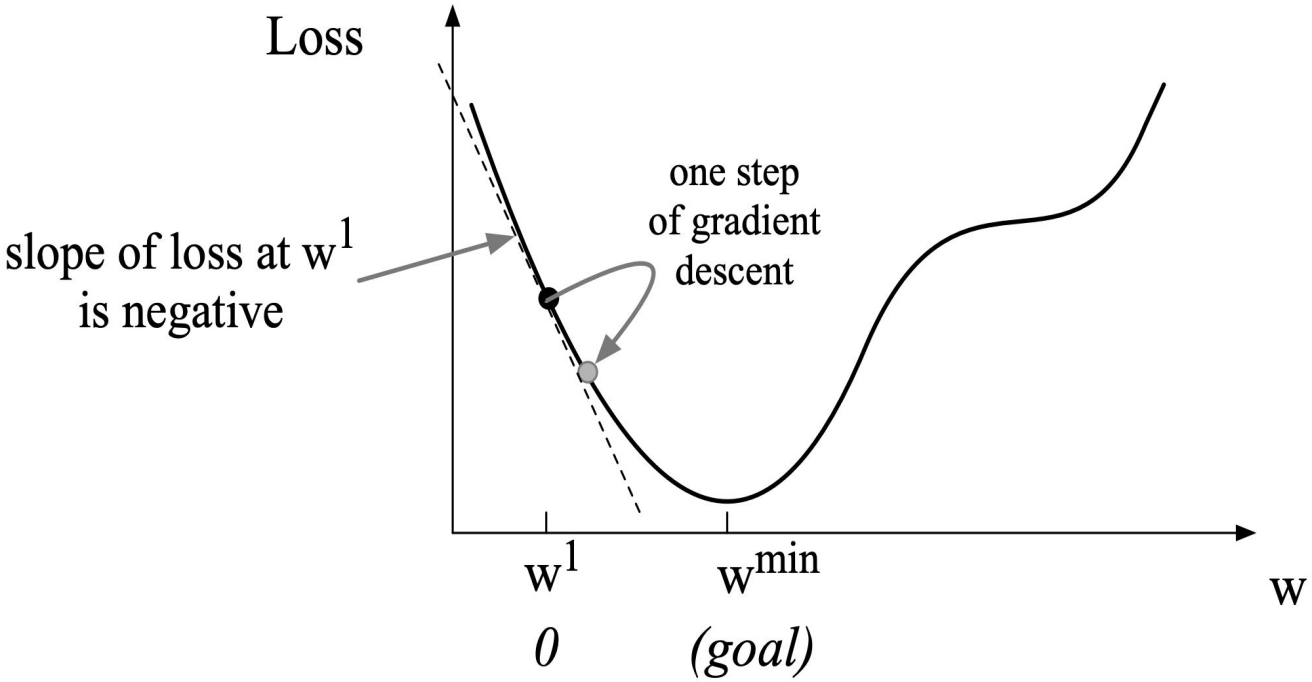
$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

Optimization algorithm: Stochastic Gradient descent

Our goal with gradient descent is to find the optimal weights: minimize the loss function we've defined for the model.

For logistic regression, this loss function is conveniently convex. A convex function has just one minimum; there are no local minima to get stuck in, so gradient descent starting from any point is guaranteed to find the minimum





Stochastic Gradient Descent

Stochastic gradient descent is an algorithm that minimizes the loss function by computing its gradient after each training example, and nudging θ in the right direction (the opposite direction of the gradient).

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ

```
    # where: L is the loss function
    #        f is a function parameterized by θ
    #        x is the set of training inputs x(1), x(2),..., x(n)
    #        y is the set of training outputs (labels) y(1), y(2),..., y(n)

θ ← 0
repeat til done    # see caption
  For each training tuple (x(i), y(i)) (in random order)
    1. Optional (for reporting):          # How are we doing on this tuple?
       Compute ŷ(i) = f(x(i); θ)          # What is our estimated output ŷ?
       Compute the loss L(ŷ(i), y(i))     # How far off is ŷ(i)) from the true output y(i)?
    2. g ← ∇θ L(f(x(i); θ), y(i))         # How should we move θ to maximize loss?
    3. θ ← θ − η g                        # Go the other way instead
return θ
```

Naive Bayes

- Naive Bayes is a family of statistical algorithms we can make use of when doing text classification.
- One of its advantages is that you get good results when data and computational resources are limited.
- Naive Bayes is based on Bayes theorem which helps us to compute the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each individual event.

This means that any vector that represents a text will have to contain information about the probabilities of appearance of the words of the text within the texts of a given category so that the algorithm can compute the likelihood of that text's belonging to the category.

Bayes Theorem

- Conditional Probability = what's the probability that something will happen, given that something else has happened?
- Spam Example = what's the probability that this text message is spam, given that it contains the word "cash"?

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$$P(\text{spam} \mid \text{cash}) = \frac{P(\text{cash} \mid \text{spam}) \times P(\text{spam})}{P(\text{cash})}$$

- Naive Bayes assumes that each event is independent, or that it has no effect on other events. This is a naive assumption, but it provides a simplified approach.
- It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Examples of Text Classification

Sentiment Analysis

Topic Labelling

Intent Detection

Language Detection