Word Vectors and Word Senses

Shubham Gupta

November 17, 2019

1 Main Idea Recap

- Iterate through words
- For each word, predict surronding word
- Word2vec maximizes objective function by putting similar words closeby.
- **Gradient Descent**: Optimize $J(\theta)$. Move in direction of negative gradient.
- **SGD**: Sample windows of data and update gradients. Much faster and practical
- With windows, we get very sparse matrices for word vectors.
  - One window will contain much fewer words, hence a majority of words in the matrix will be 0.
  - Problem: Update only word vectors that we are seeing?
  - Solution: 2 fold
    - Sparse matrix operations i.e update only those rows with non-zero values in it
    - Hash for word vectors

2 Negative sampling

- Denominator is expensive to compute.
- Try negative sampling to reduce computation.
- **Idea**: Train binary logistic regressions for a true pair(center word and it's context words in the window) versus several noise pairs(center word paired with random word).
- $P(w) = \frac{U(w)^{\frac{3}{4}}}{Z}$
- The power above reduces the chance of sampling more common words. This was obtained by trial and error i.e trained as a hyperparameter
- Dot product and negating it is equal to taking $1 - P()$. Math trick.

- SVD explained. Used to reduce the size of the matrix.
- **Hacks**
  - Scale counts i.e $min(X, t), with \;= 100$

3 Encoding meaning

- Ratios of co-occurence probs can encode meaning components.
- If dot product = log of co-occurence prob then vector diff = log(co-occurence probs)
- Glove model is based on above method.

$$J = \sum_{n=1}^{V} (X_{ij})(w_i^T w_j + b_i + b_j - log(X_{ij})^2).$$

- Advantages
  - Fast training
  - Scalable to huge corpora
  - Good performance on small corpus and vectors as well

4 Evaluation

- Intrinsic
  - Evaluate on specific task
  - Fast compute
  - Not clear outputs on results for related tasks
- Extrinsic
  - Eval on real task
  - Long time to compute accracy
  - Difficult to diagnose results.