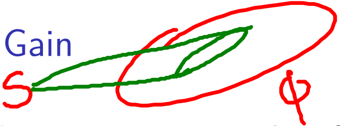
Lecture 26b: Unsupervised Learning: Dimensionality Reduction, Embeddings, PCA etc

Instructor: Prof. Ganesh Ramakrishnan

Recall: Supervised Feature Selection based on Gain



- S is a sample of training examples, p_{C_i} is proportion of examples with class C_i in S
- Entropy measures impurity of S: $H(S) \equiv \sum_{i=1}^{K} -p_{C_i} \log_2 p_{C_i}$
- Selecting R best attributes: Let $\mathcal{R} = \emptyset$
- $Gain(S, \phi_i)$ = expected **Gain** due to choice of ϕ_i Eg: Gain based on entropy -
  $Gain(S, \phi_i) \equiv H(S) \; - \sum_{v \in Values(\phi_i)} \frac{|S_v|}{|S|} H(S_v)$

  Do:
  1. $\phi^* = \underset{\phi_i \setminus \mathcal{V}}{\text{argmax}}\, Gain(S, \phi_i)$
  2. $\mathcal{R} = \mathcal{R} \cup \{\phi^*\}$

  Until $|\mathcal{R}| = R$

} Greedily select next feature that maximizes gain

Q: Other measures of **Gain**: Gini Index, Classification Error, *etc.*

Supervised Feature Subset Selection (Optional)

- One can also Optimally select ~~subset of features~~ using Iterative Hard Thresholding[1] for **Optimal Feature Selection**
- **Input:** Error function $\mathbf{E}(\mathbf{w})$ with gradient oracle to compute $\nabla\mathbf{E}(\mathbf{w})$ sparsity level s, step-size η:
- $\mathbf{w}^{(0)} = 0,\ t = 1$
- while not converged do
  1. $\mathbf{w}^{(t+1)} = P_s\left(\mathbf{w}^{(t)} - \eta\nabla_{\mathbf{w}}\mathbf{E}(\mathbf{w}^{(t)})\right)$ //Projection function $P_s(.)$ picks the highest weighted s features as per the update $\mathbf{w}^{(t)} - \eta\nabla_{\mathbf{w}}\mathbf{E}(\mathbf{w}^{(t)})$ and sets rest to 0
  2. $t = t + 1$
- end while
- Output: $\mathbf{w}^{(t)}$

[1] https://arxiv.org/pdf/1410.5137v2.pdf

From Supervised to Unsupervised Dimensionality Reduction

Recap: Feature Selection

- Supervised (greedy) Feature Selection based on Gain
- Optimally feature subset Selection (Eg: Lasso or Iterative Hard Thresholding)

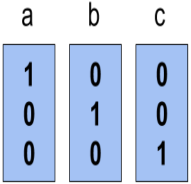
Question: What if one wants to do dimensionality reduction independent of any class-based supervision?

Recap: One Hot Encoding for Characters

- With 3 characters in vocabulary, *a*,*b* and *c*, what would be the best encoding to inform each character occurrence to the network?
- One Hot Encoding: Give a unique key k to each character in alpha-numeric order, and encode each character with a vector of vocabulary size, with a 1 for the k^{th} element, and 0 for all other elements.

Recap: One Hot Encoding for Characters

- With 3 characters in vocabulary, a,b and c, what would be the best encoding to inform each character occurrence to the network?
- One Hot Encoding: Give a unique key k to each character in alpha-numeric order, and encode each character with a vector of vocabulary size, with a 1 for the k^{th} element, and 0 for all other elements.



Encoding Words

In unsupervised setting, these are hidden

How to encode the words for the task of labeling a drama reviews as "liked" or "not liked" ?

- Review 1: The drama was interesting, loved the way each scene was directed. I simply loved everything in the drama.
- Review 2: I had three boring hours. Very boring to watch.
- Review 3: I liked the role each that was assigned to each super star. Especially loved the performance of actor.
- Review 4: Though I hate all the dramas of the director, this one was an exception with lot of entertainment.

Q: Learn model of interplay between word (representations) given context even ignoring class label

Encoding Words

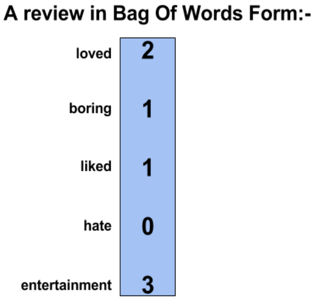
How to encode the words for the task of labeling a ''*drama*' reviews as "liked" or "not liked" ?

- One Hot Encoding of Words.
- Bag Of Words, similar to one hot encoding of characters
  - Use the vocabulary of highly frequent words in reviews.
  - Use the word frequency in each review instead of "1".

Encoding Words

How to encode the words for the task of labeling a ''*drama*' reviews as "liked" or "not liked" ?

- One Hot Encoding of Words.
- Bag Of Words, similar to one hot encoding of characters
  - Use the vocabulary of highly frequent words in reviews.
  - Use the word frequency in each review instead of "1".



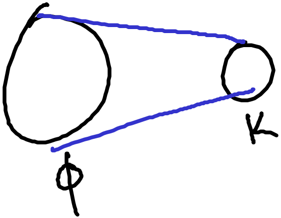
(Word) Embedding: Motivation

2 loved
3 excellent

unnecessarily separately accounting for "loved" & "excellent"

Limitations of Bag of Words or One Hot Encoding for words

- High Dimension: In real life scenario, the vocabulary size could be huge.
- Lacks Contextual Similarity - e.g. liked and loved are contextually similar words.



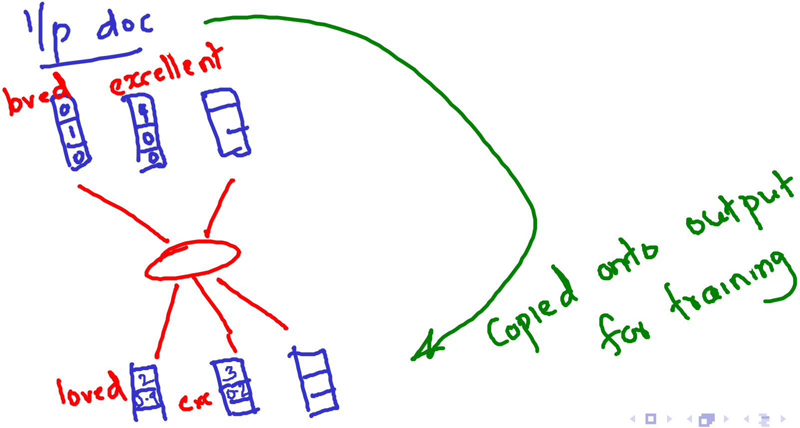
(spirit of unsupervised learning)

So far, our view of ϕ was through K classes.
Can we view ϕ & interactions directly?

(Word) Embedding: Motivation

Dimensionality Reduction techniques.

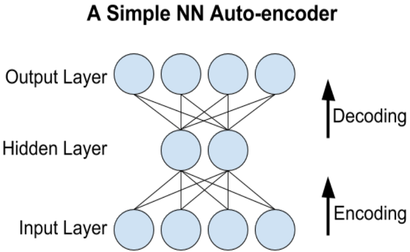
- Bag of Frequent Words: Contextual similarity is still lacking.
- **What happens if one passes a one hot encoded word as both input and output to a NN?**



(Word) Embedding: Motivation

Dimensionality Reduction techniques.

- Bag of Frequent Words: Contextual similarity is still lacking.
- **What happens if one passes a one hot encoded word as both input and output to a NN?**
- **NN Auto-encoder: Output has same form as input**. We extract the encoded vector from the hidden layer.



reviews similar to review1

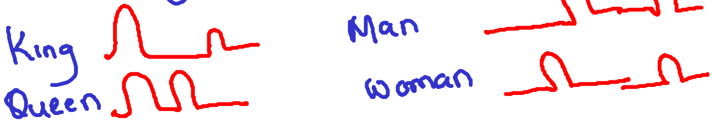
review1

(Word) Embedding: Motivation

After unsupervised training with lot of online data, can a machine answer the questions like:-

- King - Man + Woman = ?   Queen [is auto encoder capable of recovering]
- If France:Paris, then Japan:?   Tokyo ["   "]

Operations in hidden layer context:



(Word) Embedding: Motivation

A Hypothetical Word Vector Representation

	King	Queen	Woman	Princess
Royalty	0.98	0.98	0.01	0.93
Masculinity	0.98	0.04	0.02	0.02
Femininity	0.05	0.92	0.99	0.95
Age	0.7	0.6	0.5	0.2

Latent
semantic
concepts from hidden layers

(Word) Embedding: Motivation

A Hypothetical Word Vector Representation

	King	Queen	Woman	Princess
Royalty	0.98	0.98	0.01	0.93
Masculinity	0.98	0.04	0.02	0.02
Femininity	0.05	0.92	0.99	0.95
Age	0.7	0.6	0.5	0.2

- What would be the vector for Man?

(Word) Embedding: Motivation

A Hypothetical Word Vector Representation

	King	Queen	Woman	Princess
Royalty	0.98	0.98	0.01	0.93
Masculinity	0.98	0.04	0.02	0.02
Femininity	0.05	0.92	0.99	0.95
Age	0.7	0.6	0.5	0.2

- What would be the vector for Man?
- King - Man + Woman = ?

(Word) Embedding: Motivation

A good rule of thumb: Overestimate # of hidden layers / nodes & regularize

A Hypothetical Word Vector Representation

	King	Queen	Woman	Princess
Royalty	0.98	0.98	0.01	0.93
Masculinity	0.98	0.04	0.02	0.02
Femininity	0.05	0.92	0.99	0.95
Age	0.7	0.6	0.5	0.2

- What would be the vector for Man?
- King - Man + Woman = ?
- If King:Man then Queen:? → More demanding of highly representative hidden layers (feature maps)

(Word) Embedding: Motivation

A Hypothetical Word Vector Representation

	King	Queen	Woman	Princess
Royalty	0.98	0.98	0.01	0.93
Masculinity	0.98	0.04	0.02	0.02
Femininity	0.05	0.92	0.99	0.95
Age	0.7	0.6	0.5	0.2

- What would be the vector for Man? [0.01, 0.98, 0.05, 0.6]'
- King - Man + Woman = ? Queen(as vector subtraction and addition give nearly same result as the vector for Queen)
- If King:Man then Queen:? Woman(as vector differences of both pairs give nearly same results)

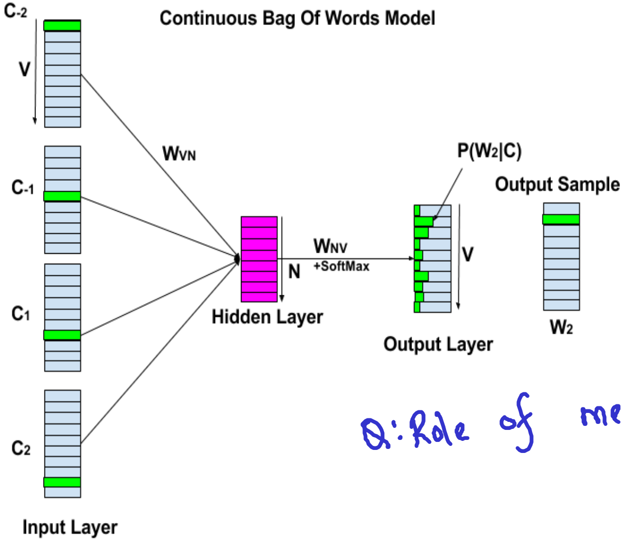
(Word) Embedding

- (Word) Embedding: Building a low-dimensional vector representation from corpus of text, which preserves the contextual similarity.
- In simple language, we want an efficient language of numbers which deep neural networks can understand as close as possible to the way we understand words.
- Training: Continuous Bag of Words Model.

(Word) Embedding

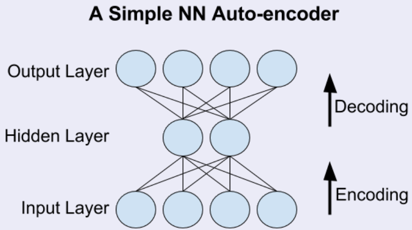
- (Word) Embedding: Building a low-dimensional vector representation from corpus of text, which preserves the contextual similarity.
- In simple language, we want an efficient language of numbers which deep neural networks can understand as close as possible to the way we understand words.
- Training: Continuous Bag of Words Model.
  - Take words in one hot encoded form. Take top V frequent words to represent each word.
  - Consider the sentence, "... I really liked the *drama*....".
  - Take a N (say 5) word window around each word and train the Neural Network with context words set $\mathcal{C}$ as input and the central word w as output.
  - For the example above use C = {"I", "really", "the" "drama"} as input and W = "liked" as output.

(Word) Embedding: Unsupervised Training



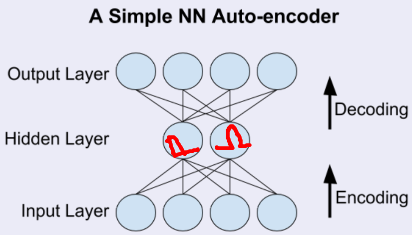
Q: Role of memory units?

What if we want Embeddings to be Orthogonal?



Hidden layer nodes
to be exclusive?

What if we want Embeddings to be Orthogonal?



- Let $\mathbf{X}$ be a random vector and Γ its covariance matrix.
- **Principal Component Analysis:** Find a rotation of the original coordinate system and express $\mathbf{X}$ in that system so that each new coordinate expresses as much as possible of the variability in $\mathbf{X}$ as can be expressed by a linear combination of the *n* entries of $\mathbf{X}$. This has application in data transformation, feature discovery, feature selection and so on.

Embeddings as Generalization of PCA

- Let $\mathbf{X}$ be a random vector and Γ its covariance matrix. Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be the n (normalized) eigenvectors of Γ.
- The n principal components of $\mathbf{X}$ are said to be $\mathbf{e}_1^T\mathbf{X}$, $\mathbf{e}_2^T\mathbf{X}$, ..., $\mathbf{e}_n^T\mathbf{X}$.

1. Let $p(X_1) = \mathcal{N}(0,1)$ and $p(X_2) = \mathcal{N}(0,1)$ and $cov(X_1, X_2) = \theta$. Find all the principal components of the random vector $\mathbf{X} = [X_1, X_2]^T$. [Tutorial 10]
2. Now, let $\mathbf{Y} = \mathcal{N}(\mathbf{0}, \Sigma) \in \Re^p$ where $\Sigma = \lambda^2 I_{p\times p} + \alpha^2 ones(p,p)$ for any $\lambda, \alpha \in \Re$. Here, $I_{p\times p}$ is a $p \times p$ identity matrix while $ones(p,p)$ is a $p \times p$ matrix of 1's. Find atleast one principal component of $\mathbf{Y}$. [Tutorial 10]