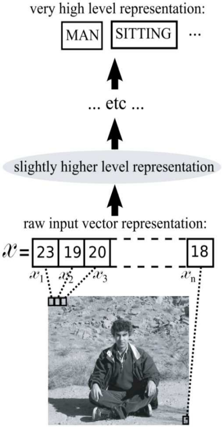
Deep Learning via Semi-Supervised Embedding

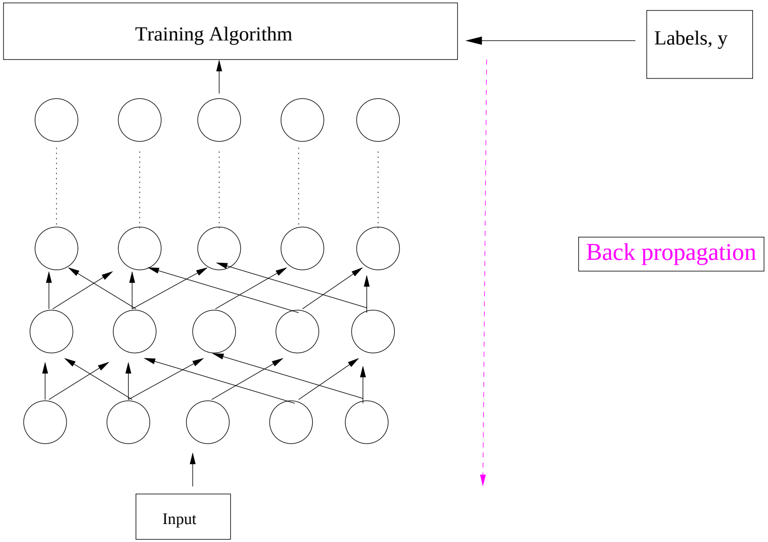
Jason Weston, Frederic Ratle and Ronan Collobert
Presented by: Janani Kalyanam

Review Deep Learning

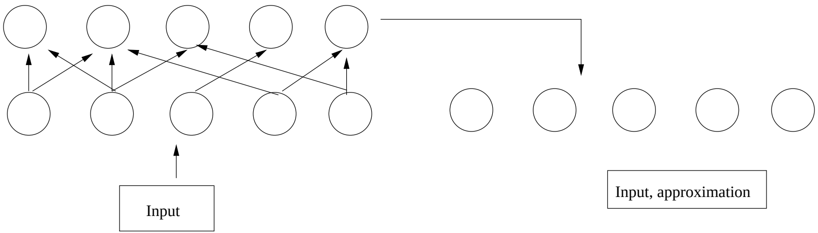
- Extract low-level features first.
- Extract more complicated features as we progress. Called pre-training.
- Perform supervised task at the end, and fine tune weights by back propagation.



Review Deep Learning, contd.



Review Deep Learning, contd.



- mimimize $||x - f_{dec}(f_{enc}(x))||^2$

Authors' point of view

- Shallow methods give nice insights to the problem, but are restrictive.
- Deep methods are complicated.
- Moreover, all the unsupervised methods proposed, like RBMs or auto-associators seem to be different from existing unsupervised learning techniques.

Authors' point of view

- Shallow methods give nice insights to the problem, but are restrictive.
- Deep methods are complicated.
- Moreover, all the unsupervised methods proposed, like RBMs or auto-associators seem to be different from existing unsupervised learning techniques.

Why not try to borrow the nice ideas from shallow methods and put it in a deep learning framework?

Proposition

- Choose an unsupervised learning algorithm *(that already exists in shallow literature)*
- Choose a model with deep architecture
- The unsupervised learning is plugged into any layer of the architecture as an auxiliary task *(as opposed to learning the unsupervised task first, and then performing fine-tuning, or back propagation)*
- Train supervised and unsupervised tasks *simultaneously*

Outline

- Review some embedding algorithms
- Embedding algorithms used in a shallow architecure
- How do the authors apply the embedding algorithm to a deep architecture?
- Experiments

Outline

- Review some embedding algorithms
- Embedding algorithms used in a shallow architecure
- How do the authors apply the embedding algorithm to a deep architecture?
- Experiments

Review: Embedding Algorithms

- General formulation

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i,j=1}^{U} L(f(x_i, \alpha), f(x_j, \alpha), W_{ij})$$

- $f(x) \in \mathbb{R}^n$ is an embedding to be learned given $x \in \mathbb{R}^d$
- L is a loss function between pairs of example
- W is a matrix of similarity/dissimilarity

Review: Embedding Algorithm (contd.)

Multidimensional Scaling

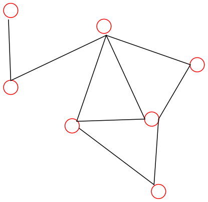
- Preserves distances between points while embedding them into a lower dimensional space

$$L(f_i, f_j, W_{ij}) = (||f_i - f_j|| - W_{ij})^2$$

Review: Embedding Algorithm contd.

Laplacian Eigen Maps

- Create a sparse, connected graph using some notion of neighbors.



- Create a weight matrix using k-nn or heat kernals: $exp^{-(x_i - x_j)/(\text{scaling})}$

Review: Embedding Algorithms contd.

- Formulation:

$$\sum_{ij} L(f_i, f_j, W_{ij}) = \sum_{ij} W_{ij}||f_i - f_j||^2$$

- Impose suitable constraints to prevent trivial solutions.

Review: Embedding Algorithm (contd.)

Margin based loss function for Siamese Networks

- Encourages similar examples to be close, and separates dissimilar ones at least by margin m

$$L(f_i, f_j, W_{ij}) = \begin{cases} ||f_i - f_j||^2 & \text{if } W_{ij} = 1 \\ max(0, m - ||f_i - f_j||^2) & \text{if } W_{ij} = 0 \end{cases}$$

Outline

- Review some embedding algorithms
- Embedding algorithms used in a shallow architecure
- How do the authors apply the embedding algorithm to a deep architecture?
- Experiments

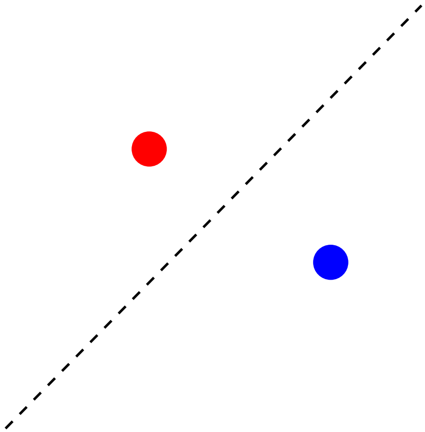
Review: Embedding in shallow architecture

Label Propagation

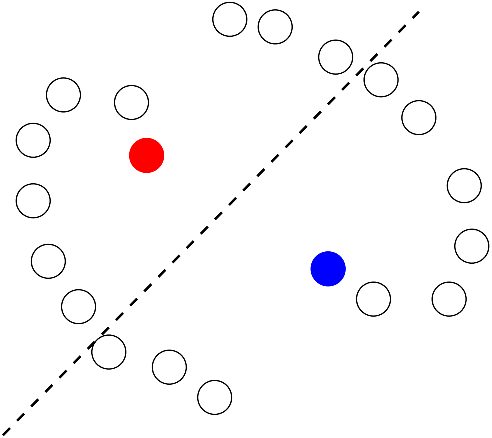
$$\min \sum_{i=1}^{L} ||f_i - y_i||^2 + \lambda \sum_{i,j=1}^{L+U} W_{ij} ||f_i - f_j||^2$$

- Encourages examples with high similarity value to get the same label
- Due to transitivity, neighbors or neighbors also get the same label

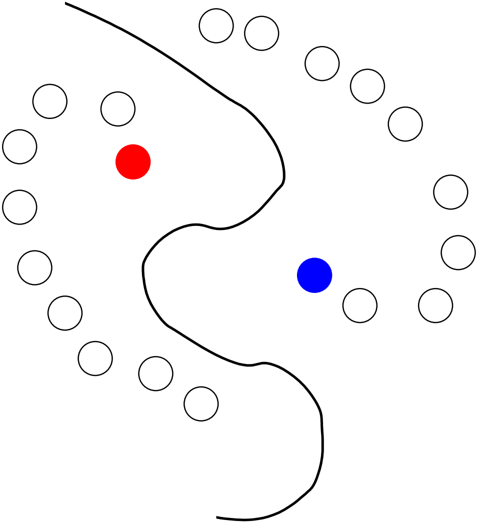
Example



Example, contd



Example, contd



Review: Embedding in shallow architecture

LapSVM

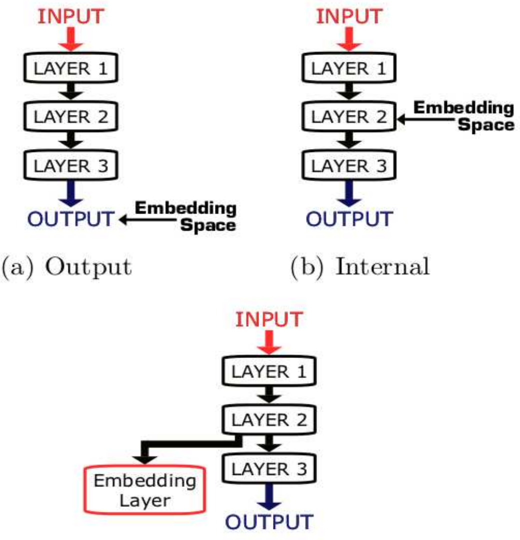
$$\min \ ||w||^2 + \gamma \sum_{i=1}^{L} H(y_i, f(x_i)) + \lambda \sum_{i,j=1}^{L+U} W_{ij} ||f(x_i) - f(x_j)||^2$$

- First two terms are from SVM formulation, third term includes unlabeled data

Outline

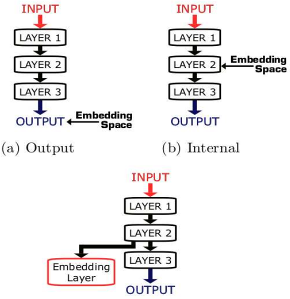
- Review some embedding algorithms
- Embedding algorithms used in a shallow architecure
- How do the authors apply the embedding algorithm to a deep architecture?
- Experiments

Semi-supervised learning in Deep architecture



(a) Output (b) Internal (c) Auxiliary

Semi-supervised learning in Deep architecture



(a) Output (b) Internal

(c) Auxiliary

- Deep learning set-up $f(x) = h^3(h^2(h^1(x)))$
- Supervised training is to minimize $L(f(x_i), y_i)$
- Can add unsupervised training to any of the layers
  - Output: $L(f(x_i), f(x_j), W_{ij})$
  - Intermediate: $L(h^2(h^1(x_i)), h^2(h^1(x_j)), W_{ij})$
  - Auxiliary: $L(e(x_i), e(x_j), W_{ij})$ where $e(x_i) = e(h^2(h^1(x)))$

Outline

- Review some embedding algorithms
- Embedding algorithms used in a shallow architecure
- How do the authors apply the embedding algorithm to a deep architecture?
- Experiments

Experiments

Small datasets

Train	g50c	Text	Uspst
SVM	8.32	18.36	23.18
TSVM	5.80	5.61	17.61
LapSVM	5.4	10.4	12.7
NN	10.6	15.7	25.1
EmbedNN	5.66	5.82	15.49

MNIST

- 2 layers, crossvalidate over the number of hidden units, and learning rate.
- W_{ij} is binary according to 10-nn criterion.

Train	1h	6h	1k	3k
SVM	23.44	8.85	7.77	4.21
TSVM	16.81	6.16	5.38	3.45
RBM	21.5	-	8.8	-
SESM	20.6	-	9.6	-
DBN-NCA	-	10.0	-	3.8
DBN-rNCA	-	8.7	-	3.3
NN	25.81	11.44	10.07	6.04
EmbedNN-O	17.05	5.7	5.7	3.59
EmbedI-1	16.86	9.44	8.52	6.02
EmbedA-1	17.17	7.56	7.89	4.93

MNIST

- 50 hidden units on each layer.
- Classical NN compared to EmbedNN-O and EmberNN-ALL
- More sophisticated experiments on video data by taking images from consecutive streams for encoding W matrix.

Train	2	4	6	8	10	15
NN	26.0	26.1	27.2	28.3	34.2	47.7
EmbedNN-O	19.7	15.1	15.1	15.0	13.7	11.8
EmbedNN-ALL	18.2	12.6	7.9	8.5	6.3	9.3

Take away..

Unsupervised learning as an auxiliary tasks seems to work well.

The End

Thank you!