Manifold Constrained Deep Neural Networks for ASR

Department of Electrical and Computer Engineering,
McGill University

Richard Rose and Vikrant Tomar

Motivation

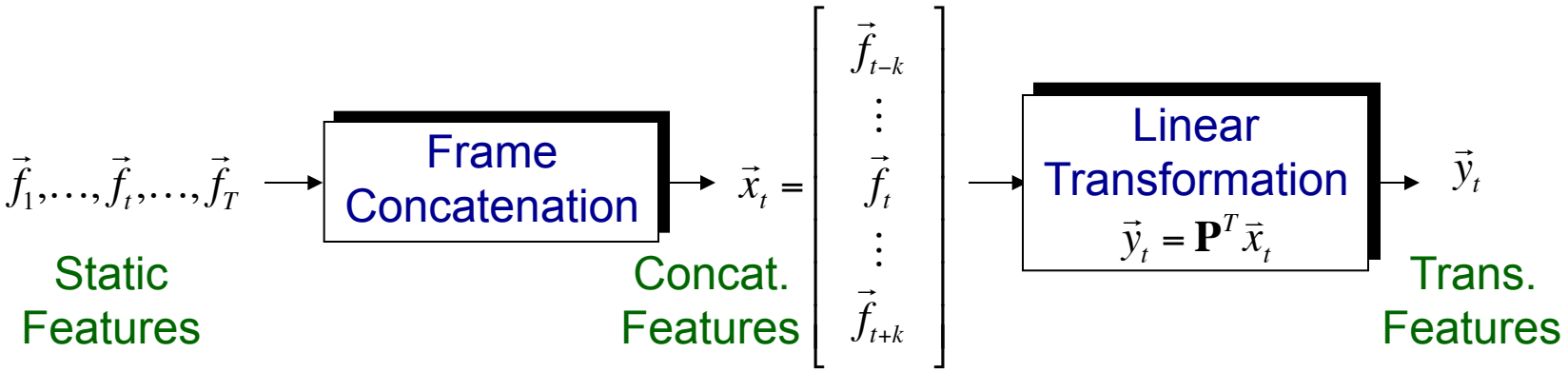
- Speech features can be characterized as lying on a low dimensional embedded manifold
- Manifolds can be highly nonlinear subspaces but have ***nice*** locally linear properties
  - Euclidean distances in local neighborhoods
  - Local curvature in differentiable manifolds
- Knowing what we don't know about speech
  - Try to know something about the manifold properties
- ***Performance Measure:*** Departure from assumptions about assumed underlying manifold
  - Analyze features to determine if local neighbor relationships or local shape assumptions are violated

Manifold Constraints in ASR

- Manifold learning and linear dimensionality reducing transformations in ASR
  - Preserving manifold based locality constraints
  - Preserving class separability along a manifold
  - Incorporating noise robust kernels
- Efficient graph embedding
  - Locality sensitive hashing (LSH): Reduce nearest neighbor computations for neighborhood graphs
- Manifold based regularization
  - Manifold regularized DNN training
  - Manifold regularized speaker adaptation
  - Regularized least squares classifier for spoken term detection

Linear Dimensionality Reducing Transforms for ASR

- Model super-segmental spectral dynamics in speech by concatenating static feature vectors:



- Dimensionality reducing linear transformation, $\mathbf{P} \in \Re^{d \times m}$ from high dimensional, m , to low dimensional, d, feature space [Soan et al, 2000]:

$$\vec{y}_t = \mathbf{P}^T \vec{x}_t$$

Optimization Criteria for Dimensionality Reduction

- Linear discriminant analysis (LDA): Class separability in a Euclidean space [Soan et al, 2000]

- Locality preserving projections (LPP): Preserve local relationships among feature vectors in the transformed space [He et al, 2002][Tang and Rose, 2008]

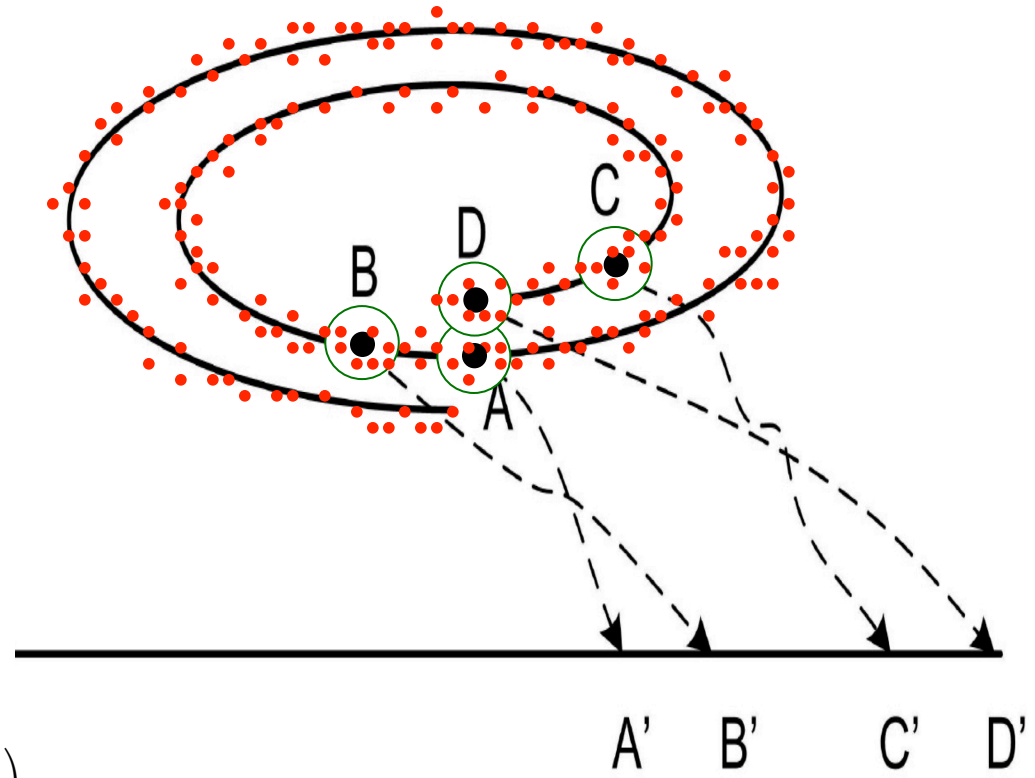
- Locality preserving discriminant analysis (LPDA): Discriminative manifold learning [Cai et al, 2007][Tomar and Rose, 2012]
  - Maximize class separability between class-specific sub-manifolds
  - Preserve local relationships on sub-manifolds

Preserving Local Relationships Along a Manifold

Assumption: High dimensional data can be considered as a set of geometrically related points resting on or close to the surface of a lower dimensional manifold

Locality: Manifold constraints among feature vectors can be applied by exploiting local class dependent neighborhood relationships $C(\vec{x}_i) = C(\vec{x}_j)$

Intrinsic Graphs: $\mathbf{G}_{int} = \{X, W_{int}\}$ Nodes are data points and weights are evaluated over nearest neighbor kernels



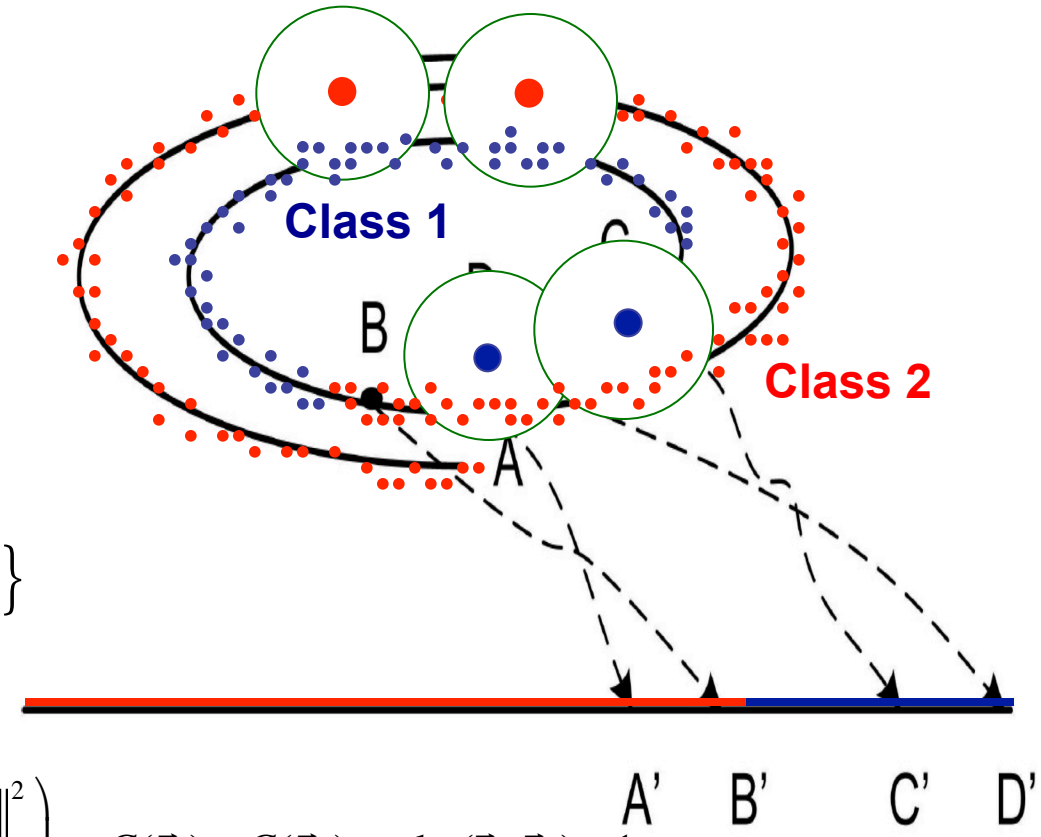
$$w_{ij} = \begin{cases} \exp\left(\dfrac{-\left\|\vec{x}_i - \vec{x}_j\right\|^2}{\rho}\right), & C(\vec{x}_i) = C(\vec{x}_j) \text{ and } e(\vec{x}_i, \vec{x}_j) = 1 \\ 0, & \text{otherwise} \end{cases}$$

Discriminative Manifold Learning

Discriminative Criteria:
Relies on a measure of discriminability across classes on a manifold

Locality:
Penalize local neighborhood relationships between data vectors across classes $C(\vec{x}_i) \neq C(\vec{x}_j)$

Penalty Graphs: $\mathbf{G}_{pen} = \{X, W_{pen}\}$
Inter-class weights are evaluated over nearest neighbor kernels:



$$w_{ij} = \begin{cases} \exp\left(\dfrac{-\left\|\vec{x}_i - \vec{x}_j\right\|^2}{\rho}\right), & C(\vec{x}_i) \neq C(\vec{x}_j) \text{ and } e(\vec{x}_i, \vec{x}_j) = 1 \\ 0, & \text{otherwise} \end{cases}$$

Discriminative Manifold Learning: Locality Preserving Discriminative Projections (LPDA)

- Define intrinsic graph, $\mathbf{G}_{int} = \{X, W_{int}\}$,and penalty graph, $\mathbf{G}_{pen} = \{X, W_{pen}\}$
- Identify matrix, $\mathbf{P}_{lpda}$, where $\vec{y}_t = \mathbf{P}_{lpda}^T \vec{x}_t$, to maximize …

$$F(\mathbf{P}_{lpda}) = \frac{F_{pen}(\mathbf{P})}{F_{\text{int}}(\mathbf{P})} = \frac{P^T X(D_{pen} - W_{pen})X^T P}{P^T X(D_{\text{int}} - W_{\text{int}})X^T P}$$

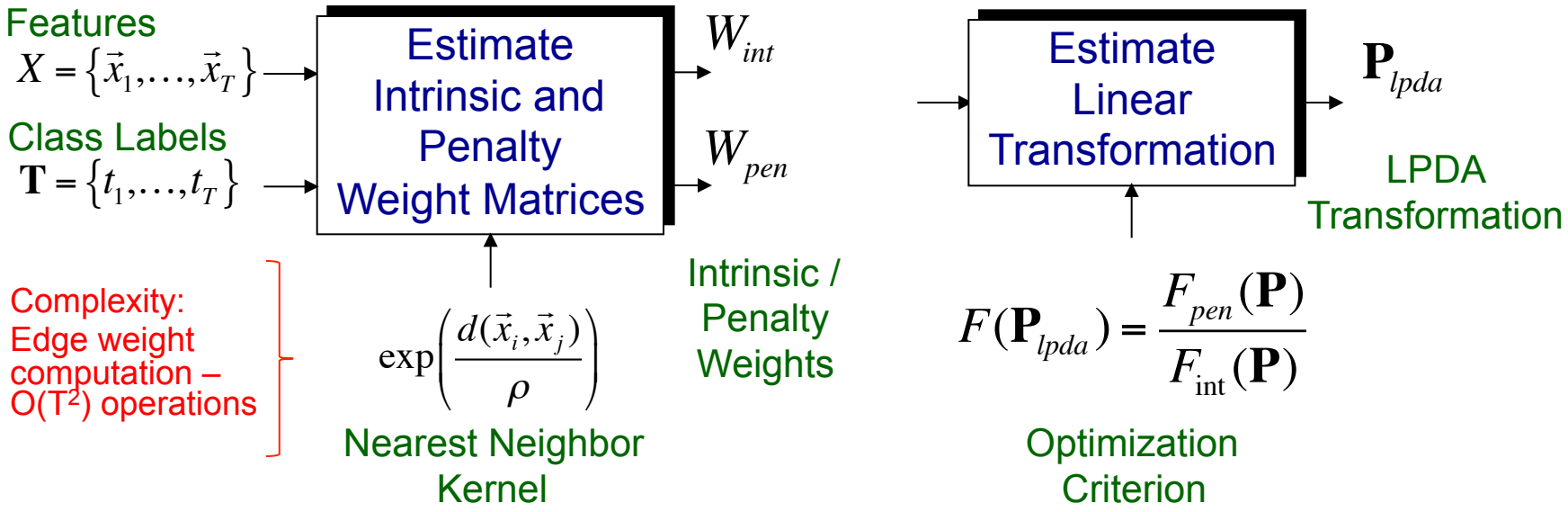
- Optimum $\mathbf{P}_{lpda}$ is the solution to generalized eigen equation [Cai et al, 2007]

$$X(D_{pen} - W_{pen})X^T p_{lpda}^j = \lambda X(D_{int} - W_{int})X^T p_{lpda}^j$$

… where p_{lpda}^j are columns of $\mathbf{P}_{lpda}$

Discriminative Manifold Learning

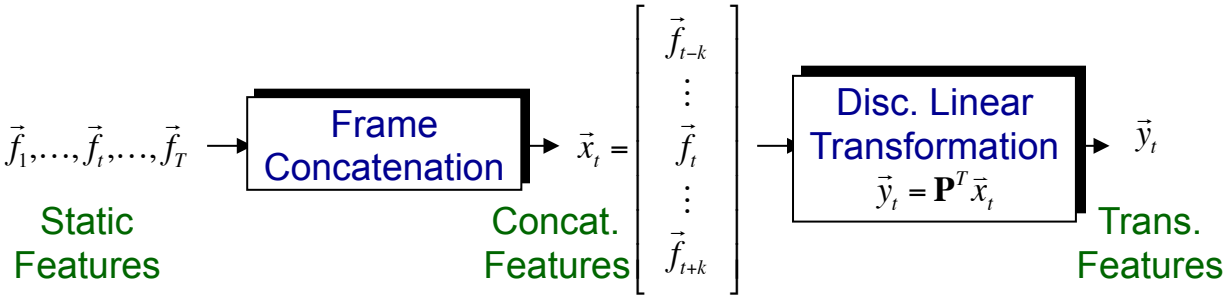
- Manifold constraints for discriminative linear projection matrix:



- Intrinsic / Penalty Weights:
  - Intrinsic Graph – Within-class neighborhoods
  - Penalty Graph – Between-class neighborhoods
- Optimization Criterion:
  - Preserve locality in the transformed space
  - Maintains class separability

Performance Evaluation – Manifold-Based LPDA

- Estimate dimensionality reducing transformations using LPDA and LDA criteria and compare ASR performance on speech-in-noise task



- Discriminative Transformation $\vec{y}_t = \mathbf{P}^T \vec{x}_t$:
  - Classes: 121 monophone clustered states
  - $\vec{x}_t$ Input dimensionality: 117, $\vec{y}_t$ Output dimensionality: 39
- Aurora-4 Task: Noise corrupted WSJ utterances [Parthar and Picone, 2002]
  - Training (14 hours): Mixed noise recordings
  - Test: Six sets of utterances from six noise types at noise levels ranging from 5 dB to 20 dB SNR

Performance Evaluation – Manifold-Based LPDA

ASR Performance on Aurora-4 Task WER (WER Reduction)			
Noise Type	Technique		
	Baseline	LDA	LPDA
Clean	15.34	15.09	13.97 (7.44)
Car	15.90	16.34	14.53 (11.08)
Babble	26.62	25.37	21.56 (15.02)
Restaurant	28.28	28.77	24.51 (14.81)
Street	31.59	29.87	27.46 (8.07)
Airport	23.65	23.65	18.96 (19.83)
Train Stn.	32.08	29.96	28.60 (4.54)
Average	**24.78**	**24.15**	**21.37 (11.51)**

- Good: LPDA reduces WER by as much as 20% relative to LDA
- Bad: LPDA requires much higher computational complexity than LDA

Manifold Constraints in ASR

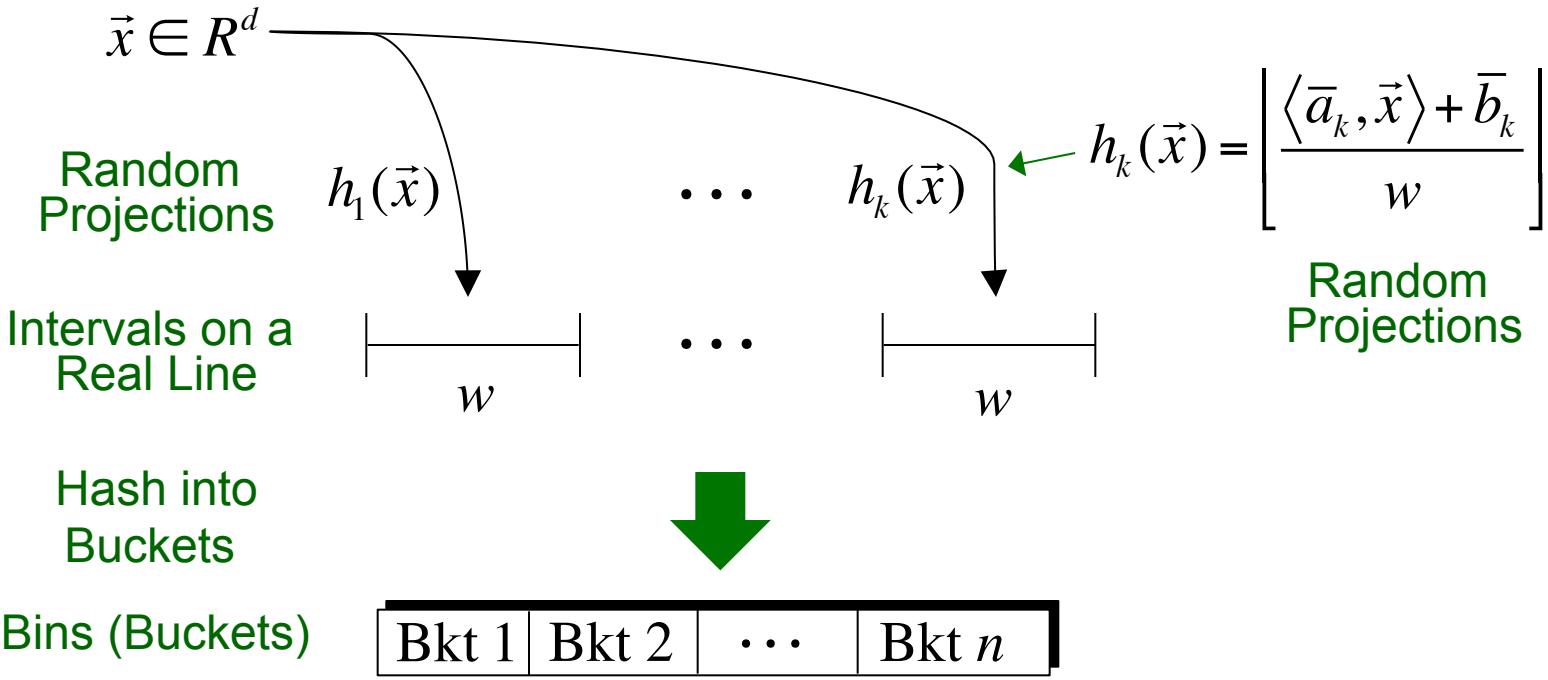
- Manifold learning and linear dimensionality reducing transformations in ASR
  - Preserving manifold based locality constraints
  - Preserving class separability along a manifold
  - Incorporating noise robust kernels
- Efficient graph embedding
  - Locality sensitive hashing (LSH): Reduce nearest neighbor computations for neighborhood graphs
- Manifold based regularization
  - Manifold regularized DNN training
  - Manifold regularized speaker adaptation
  - Regularized least squares classifier for spoken term detection

Locality Sensitive Hashing – Reducing Complexity of Manifold Based Techniques

- **Problem:** Estimating Affinity weight matrices $W = \left[w_{i,j} \right]_{T \times T}$ requires computational complexity of $O\left(T^2\right)$
  - T ranges from ~1 M to ~1 B frames for speech training corpora
- **Locality Sensitive Hashing (LSH):** A randomized algorithm for hashing vectors into bins such that adjacent vectors are more likely to fall into the same bin [Pauleve et al, 2010][Datar et al, 2004][Jansen et al, 2011][Tomar and Rose, 2013]
- **Complexity Reduction:** Apply LSH to fast computation of neighborhood graphs
- **Goal:** Reduce complexity with minimum impact on ASR performance

Locality Sensitive Hashing – Creating Hash Tables

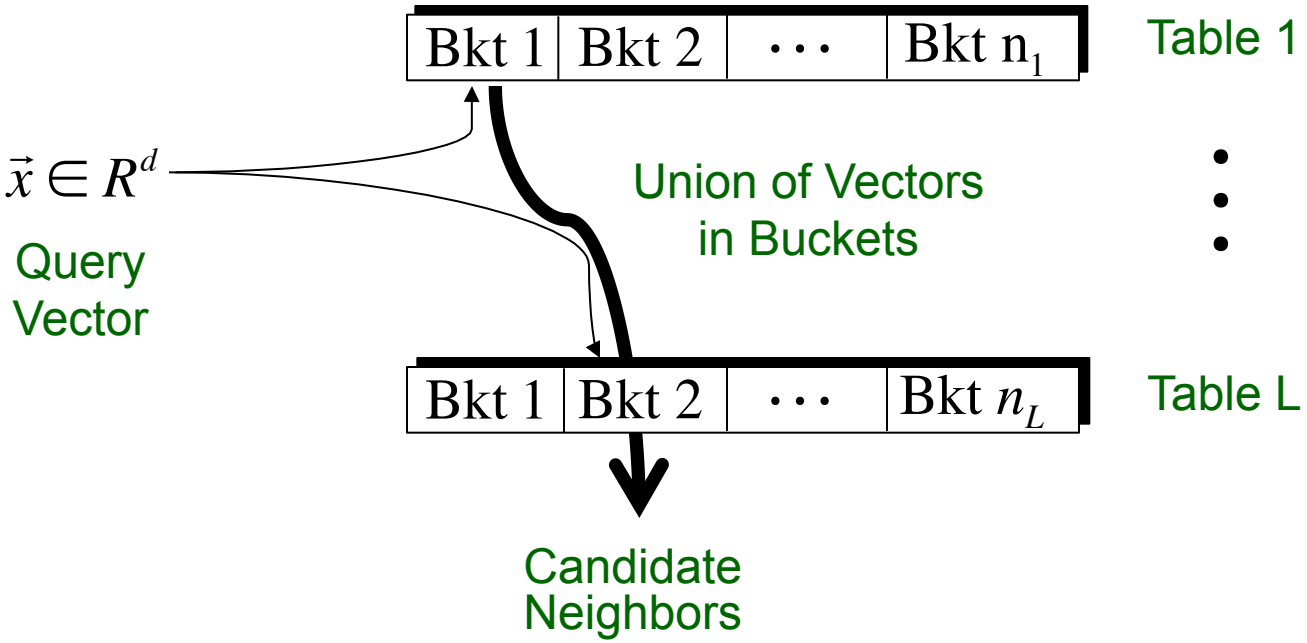
- Hash vectors to integer values or "buckets" using random projections:



- Multiple Hash Tables: Increase probability of finding nearest neighbor:

Locality Sensitive Hashing – Nearest Neighbor Search

- Hash query point to a bucket in each of L tables



- Obtain candidate K-nearest neighbors for $\vec{x}$ from union of vectors assigned to buckets

LSH – Complexity vs. Performance for Estimating Affinity Matrices

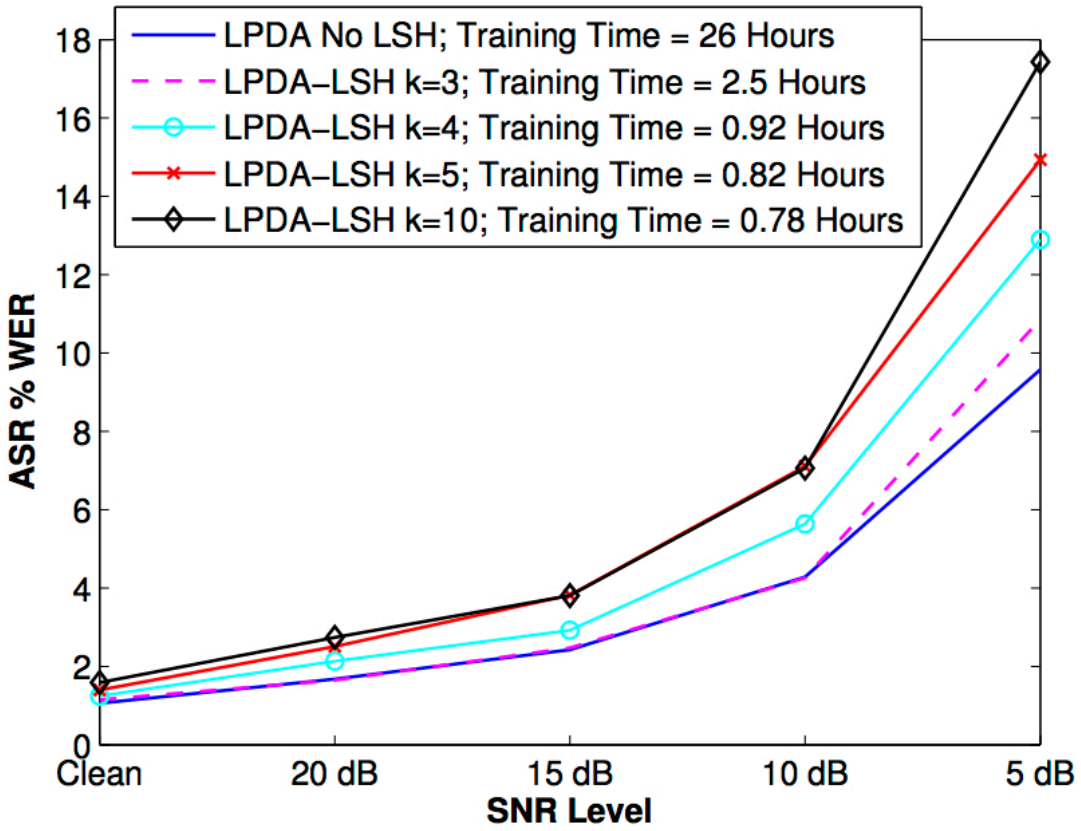
Task Domain

Aurora 2: 3 noise types
Training Data: 8440 utt.
Training Frames: T = 1.47 M

LSH Parameterization

Vector dimension: d=117
Nearest Neighbors: K = 200
Tables: L=6
Projections: k = 3,…,10

Ave. WER for 3 Noise Types on Aurora 2 Task



- Order of magnitude reduction in complexity with negligible impact on ASR word error rate

Manifold Constraints in ASR

- Manifold learning and linear dimensionality reducing transformations in ASR
  - Preserving manifold based locality constraints
  - Preserving class separability along a manifold
  - Incorporating noise robust kernels
- Efficient graph embedding
  - Locality sensitive hashing (LSH): Reduce nearest neighbor computations for neighborhood graphs
- Manifold based regularization
  - Manifold regularized DNN training

Manifold Regularized Deep Neural Network Training

- Lots of recent work dealing with local optima in DNN training [Glorot and Bengio, 2010]
  - Layer-by-layer generative pre-training [Dahl et al, 2012]
  - Discriminative pre-training [Seide et al., 2011]
  - Rectified Linear Units (ReLU) and Drop-out [Dahl et al, 2013]
- Feature spaces with strong local constraints lead to improved learning for DNNs [Mohamed et al., 2012]
  - Manifold learning can be used to “enforce” locality constraints
- Manifold regularization has already been applied in other applications
  - Learning in multilayer perceptrons (MLPs) [Weston et al, 2008]
  - Regularized least squares (RLS) classifiers [Belkin et al, 2006]

Manifold Regularization for DNN Training

- Motivation:
  - Enforce explicit locality constraints in hidden layers of DNN
  - Provide alternative to pre-training based regularization strategies

- Locality Constraints - Modified cross-entropy objective function:

$$Q(\theta) = \sum_{i=1}^{T} \left\{ V_\theta(y_i, t_i) + \gamma \sum_{j=1}^{K} \omega_{ij}^{\text{int}} \left\| y_i - y_j \right\|^2 + \kappa \left\| \theta \right\| \right\}$$

Cross Entropy | Manifold Constraints | Norm Regularizer

- Locality / Discriminative Constraints:

$$Q(\theta) = \sum_{i=1}^{T} \left\{ V_\theta(y_i, t_i) + \gamma \sum_{j=1}^{K} \omega_{ij}^{\text{int}} \left\| y_i - y_j \right\|^2 + \xi \sum_{j=1}^{K} \omega_{ij}^{pen} \left\| y_i - y_j \right\|^2 + \kappa \left\| \theta \right\| \right\}$$

Cross Entropy | Intrinsic Manifold | Penalty Manifold | Norm

Implications of Manifold Regularized DNN Training

- Constrains the DNN outputs to lie along an embedded manifold
  - Preserves local relationships among speech feature vectors at the output of the DNN
- Increase in computational cost
  - High cost of computing nearest neighbors - locality sensitive hashing
  - High cost of computing gradients of manifold regularized objective function to incorporate nearest neighbors

Performance – Manifold Regularization DNN

- Aurora-2 Task: Noise corrupted digit utterances (~4 hours training)
  - **MFCC:** Baseline mel-frequency cepstrum coefficients
  - **DNN:** DNN with no pre-training or regularization
  - **MRDNN:** Manifold Regularized DNN

Ave. WER Over 4 Noise Types for Aurora-2						
Conditions	Features	Noise Level - SNR (dB)				
		Clean	20	15	10	5
Subway, Exhibition, Babble, and Car Noise Types	**MFCC**	1.87	3.10	3.89	6.57	13.75
	DNN	0.98	1.17	1.87	3.17	7.65
	MRDNN	**0.71**	**0.98**	**1.52**	**2.67**	**6.73**

- WER Reduction: Consistent (~20%) reduction relative to unregularized DNN over a range of SNRs

- Complexity: Computation for back propagation weight estimation increased by a factor of K (number of nearest neighbors)

Performance: Manifold Regularized DNN

- Aurora-4: Noise corrupted WSJ utterances [Parthar and Picone, 2002]
  - Training (14 hours): Mixed noise recordings
  - Test: Six sets of utterances from six noise types at noise levels ranging from 5 dB to 20 dB SNR

ASR Performance on Aurora-4 Task WER (WER Reduction)			
Noise Type	Technique		
	MFCC	**DNN**	**MRDNN**
Clean	13.10	9.10	**8.07**
Car	14.44	11.45	**10.63**
Babble	25.63	18.23	**17.52**
Restaurant	27.26	22.10	**20.96**
Street	30.62	21.15	**20.87**
Airport	22.59	17.43	**16.87**
Train Stn.	31.23	21.91	**20.12**

- Consistent WER reduction relative to unregularized DNN over a range of noise types

Implications of Manifold Regularized DNN Training

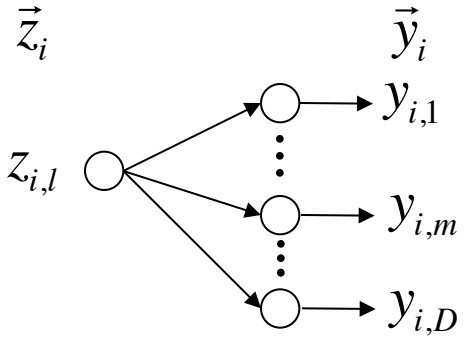
- Constrains the DNN outputs to lie along an embedded manifold
  - Preserves local relationships among speech feature vectors at the output of the DNN
- Increase in computational cost
  - High cost of computing nearest neighbors - locality sensitive hashing
  - High cost of computing gradients of manifold regularized objective function to incorporate nearest neighbors

Review: Parameter Updates in DNN Training

- Gradient is estimated for cross entropy objective function:

$$Q(\theta) = \sum_{i=1}^{T} \left\{ V_\theta \left(\vec{y}_i, \vec{t}_i \right) \right\} \longrightarrow \nabla_{\theta_{l,m}} Q(\theta) = -\left(y_{i,m} - t_{i,m} \right) z_{i,l}$$

… where outputs, $\vec{y}_i$, are obtained from inputs, $\vec{z}_i$, by forward propagation:

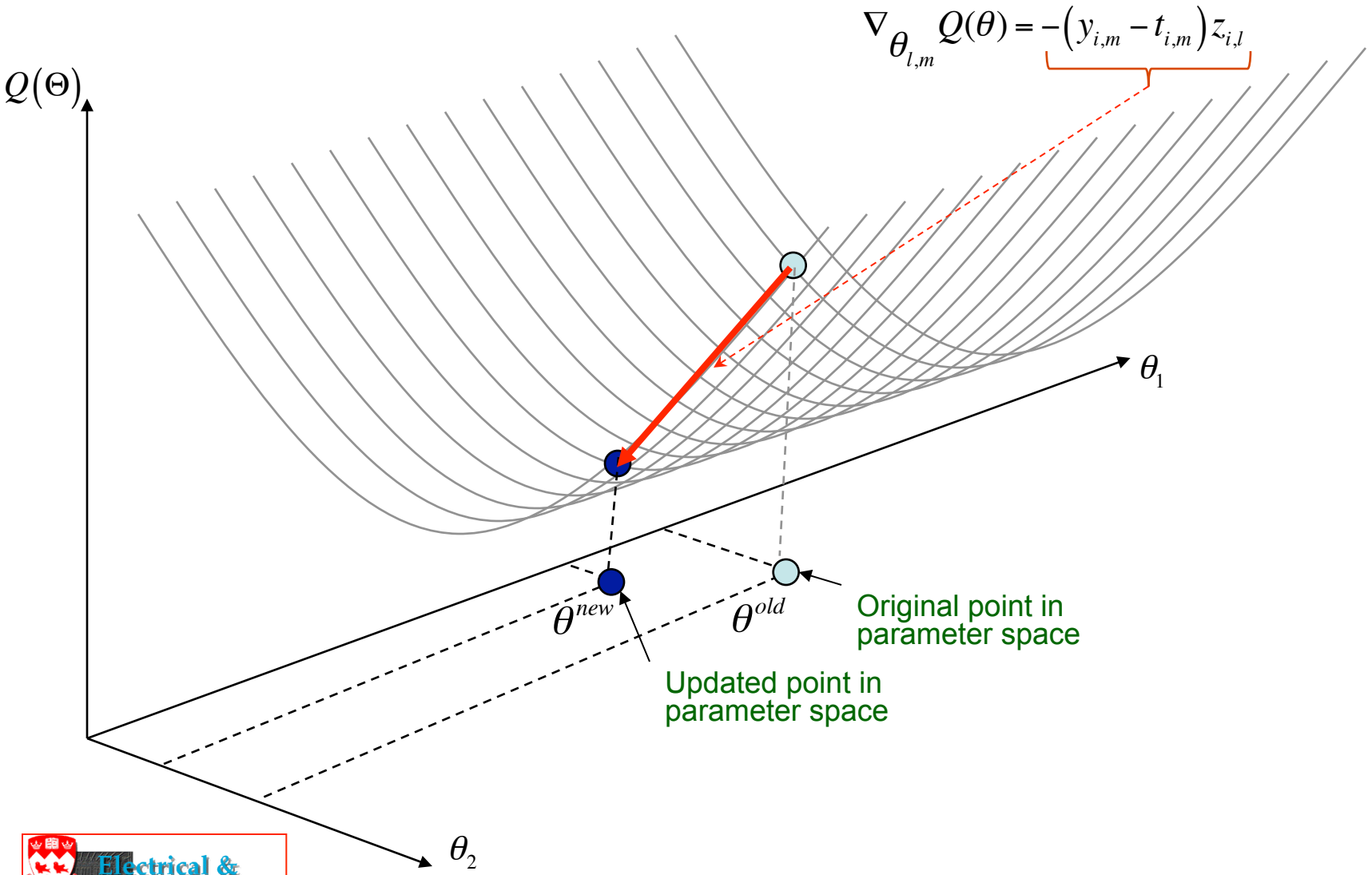


$$y_{i,m} = f\left(\sum_l \theta_{m,l} z_{i,l} \right)$$

… and parameters are updated by gradient descent:

$$\theta_{l,m}^{new} = \theta_{l,m}^{old} + \eta \nabla_{\theta_{l,m}} Q(\theta)$$

Review: Parameter Updates in DNN Training



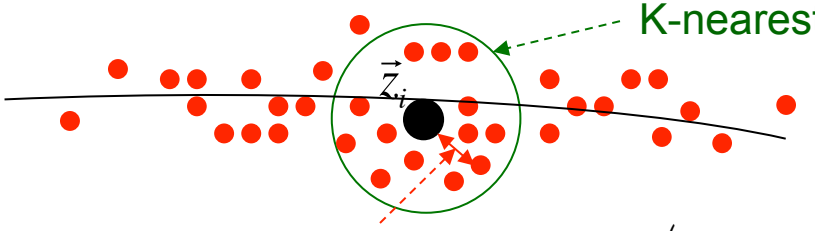
Parameter Updates in MRDNN Training

- Estimate gradient for manifold regularized objective function:

$$Q(\theta)=\sum_{i=1}^{T}\left\{V_\theta\left(y_i,t_i\right)+\gamma\sum_{j=1}^{K}w_{ij}^{\text{int}}\left\|y_i-y_j\right\|^2\right\} \longrightarrow$$

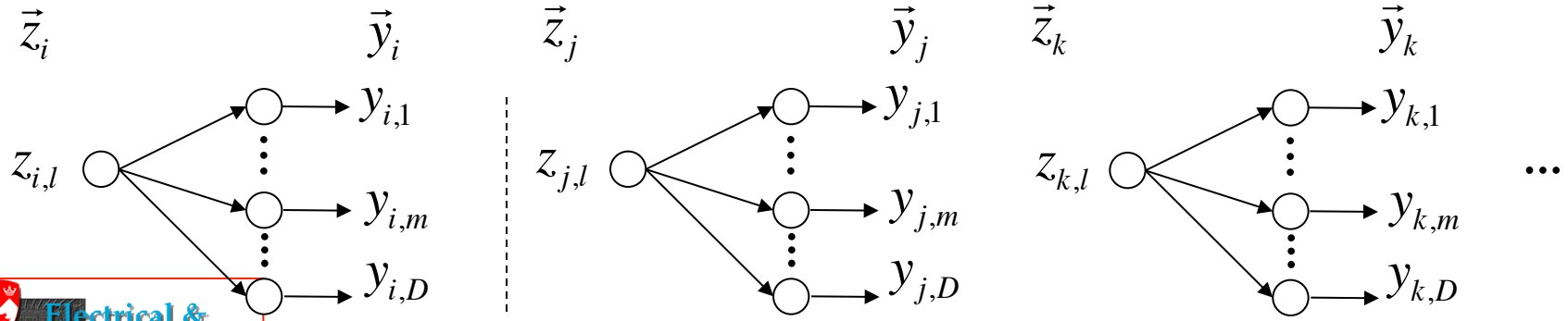
$$\nabla_{\theta_{l,m}}Q(\theta)=-\left(y_{i,m}-t_{i,m}\right)z_{i,l}-\frac{2\gamma}{K}\sum_{j=1}^{K}\omega_{i,j}\left(y_{i,m}-y_{j,m}\right)\left(\frac{\partial y_{i,m}}{\partial\theta_{l,m}}-\frac{\partial y_{j,m}}{\partial\theta_{l,m}}\right)$$

- **Prior to DNN Training**: Estimate manifold affinity matrix weights:



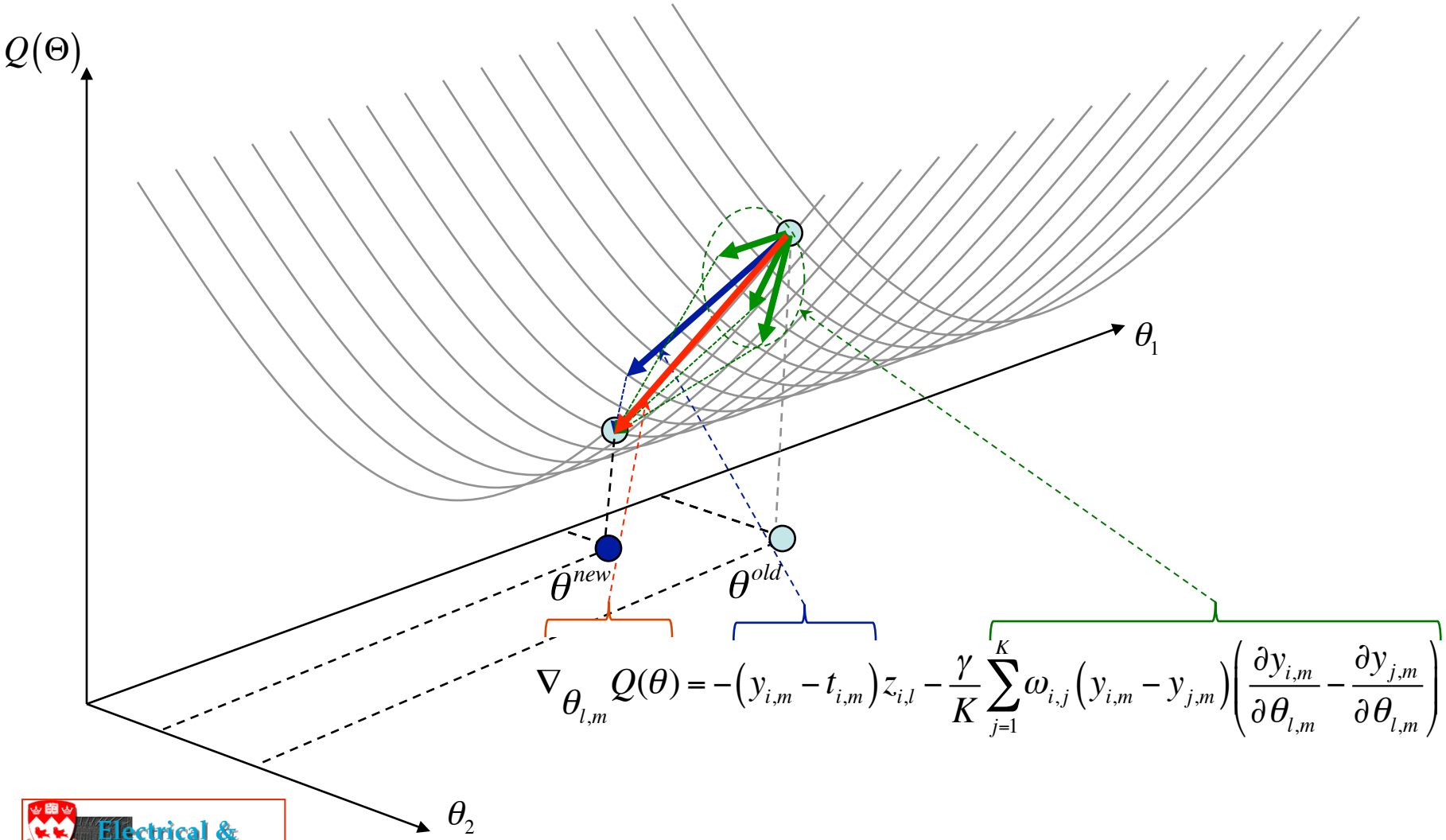
Affinity weights: $\omega_{ij}=\exp\left(-\left\|\vec{z}_i-\vec{z}_j\right\|^2/\rho\right)$

- **During DNN Training**: Forward Propagate $\vec{z}_i$ and its K NNs $\{\vec{z}_j : e(\vec{z}_i,\vec{z}_j)=1\}$:



Parameter Updates in MRDNN Training

- K nearest neighbors of $\vec{z}_i$ contribute to estimate of gradient:



Comparisons with Other DNN Regularization Approaches

- Generative Pre-training of DNNs
  - Has shown no improvement in our task domains over random network initialization
- Rectified Linear Units (ReLU): $f(x) = \max(0, x)$
  - Replace logistic non-linear units in DNN
  - Current work is to apply manifold regularization in ReLU networks
- L2 Parameter Norm Regularization - Modified cross-entropy objective function:

$$Q(\theta) = \sum_{i=1}^{T} \left\{ V_\theta\left(y_i, t_i\right) + \kappa \left\| \theta \right\| \right\}$$

Cross Entropy

Norm Regularizer

… Found to have limited impact on WER for our task

Performance – L2 Regularized DNN

- Results: Aurora-2 Task (~4 hours training)
  - **DNN:** DNN with no pre-training or regularization
  - **MRDNN:** Manifold Regularized DNN

Ave. WER Over 4 Noise Types for Aurora-2						
Conditions	Features	Noise Level - SNR (dB)				
		Clean	20	15	10	5
Subway, Exhibition, Babble, and Car Noise Types	MFCC	1.87	3.10	3.89	6.57	13.75
	DNN	0.98	1.17	1.87	3.17	7.65
	MRDNN	**0.71**	**0.98**	**1.52**	**2.67**	**6.73**
	L2-DNN	0.91	1.16	1.63	3.01	7.03

… **L2-DNN**: L2 Norm Regularized DNN

Summary

- **Discriminative Manifold Learning:** Up to 20% reduction in WER relative to LDA for noise corrupted Wall Street Journal Utterances (Aurora 4)
- **Efficient Graph Embedding:** Using locality sensitive hashing (LSH) for estimating neighborhood graphs provides an order of magnitude reduction in computation time with negligible impact on WER (Aurora 2)
- **Manifold Based Regularization:** Preliminary results show that manifold regularization of DNN training for tandem bottleneck features reduces WER by up to 30% - 40% (Aurora 2)