Reviews of DeepLDA

Presenter: Sarah Kim

2018.12.28

Linear Discriminant Analysis

- Let $x_1, \ldots, x_N = X \in \mathbb{R}^{N \times p}$ denote a set of N samples belonging to C different classes $c \in \{1, \ldots, C\}$.
- Let

$$\bar{x} = \frac{1}{N} \sum_i x_i, \quad m_c = \frac{1}{N_c} \sum_{i \in c} x_i,$$

  where $N_c = \#\{i \in c\}$.
- LDA finds a linear combination $a^\top x_i$ s.t. the between class variance is maximized relative to the within-class variance:

$$\max_a \frac{a^\top S_B a}{a^\top S_W a}, \tag{1}$$

  where $S_B = \sum_c N_c(m_c - \bar{x})(m_c - \bar{x})^\top$, $S_W = \sum_c \sum_{i \in c} (x_i - m_c)(x_i - m_c)^\top$

Generalization of Linear Discriminant Analysis

- X_c are the observations of class c and m_c is the per-class mean vector.
- LDA finds a linear projection $A \in \mathbb{R}^{r \times p}, r < p$ s.t.

$$\underset{A}{\operatorname{argmax}} \frac{|AS_BA^\top|}{|AS_WA^\top|}, \tag{2}$$

  where S_B, S_W are the between, within scatter matrices.
- A in Eq. (2) can be obtained by the eigenvectors corresponding to the r largest eigenvalues of

$$S_B e_i = v_i S_W e_i, \quad i = 1, \ldots, r \tag{3}$$

Feature Extraction with Deep Neural Networks by a Generalized Discriminant Analysis

Stuhlsatz, A., Lippel, J., & Zielke, T. (2012)

IEEE transactions on neural networks and learning systems

Introduction

- The generalized discriminant analysis (GerDA) is a generalization of the classical LDA on the basis of DNNs.
- LDA often fails in real-world applications, since a linear mapping A cannot transform arbitrarily distributed r.v.s into independently Gaussian.
- Main idea

  Find nonlinear mapping $f\colon \mathbb{R}^p \to \mathbb{R}^r$ s.t.

$$\max_f \operatorname{trace}(S_T^{-1} S_B),$$

  where S_T and S_B defined on $h = f(x)$.

Generalized Discriminant Analysis

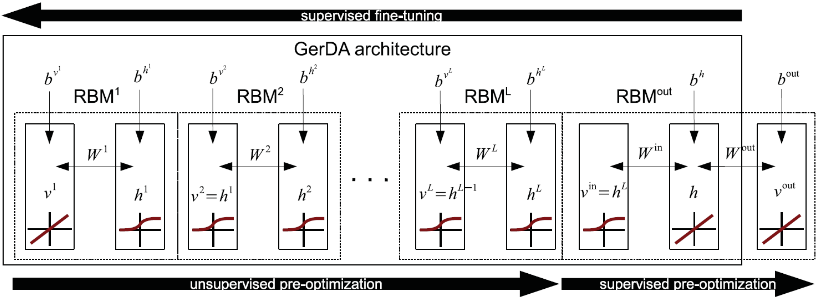


Figure : GerDA architecture

Generalized Discriminant Analysis

- Note that the objective function $\mathrm{trace}(S_T^{-1} S_B)$ overemphasizes large distances of between-class variation.
- GerDA is fine-tuned by maximizing

$$\mathrm{trace}((S_T^{\delta})^{-1} S_B^{\delta}),$$

where $S_T^{\delta} := S_W + S_B^{\delta}$ and

$$S_B^{\delta} := \frac{1}{2N^2} \sum_{i,j=1}^{C} N_i N_j \times \delta_{i,j} \times (m_i - m_j)(m_i - m_j)^{\top}$$

$$\delta_{i,j} := \begin{cases} 1/\|m_i - m_j\|^2 & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}$$

Visualization Results

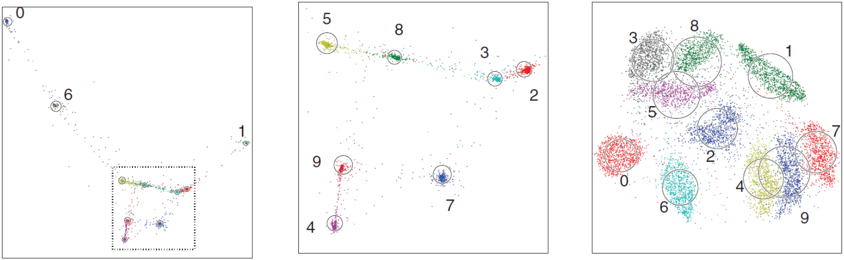


Figure : Comparison of 2-D mappings obtained using GerDa, t-SNE on the MNIST test images

Appendix: Pre-Optimization

- Unsupervised training of a single binary RBM of the ith layer $(2 \leq i \leq L)$ is performed via s.g.d. in the KL divergence

$$d(P^0 \| P^\infty; \Theta^i) := \sum_{v^i} P^0(v^i) \log \left(\frac{P^0(v^i)}{P^\infty(v^i; \Theta^i)} \right)$$

assumming $s^i := ((v^i)^\top, (h^i)^\top)^\top$, $v^i \in \{0,1\}^{N_{v^i}}$, $h^i \in \{0,1\}^{N_{h^i}}$ with distribution

$$P^\infty(v^i; \Theta^i) = \frac{1}{Z(\Theta^i)} \sum_{h^i} \exp\Big(- H(s^i; \Theta^i) \Big)$$
$$Z(\Theta^i) := \sum_{s^i} \Big(- H(s^i; \Theta^i) \Big)$$

given the network parameters $\Theta^i := (W^i, b^i)$.

Appendix: Pre-Optimization

- For binary states,

$$H(s^i;\Theta^i) := -(v^i)^\top W^i h^i - (b^i)^\top s^i$$

- Since v^1 of an input layer RBM are modeled continuously and Gaussian-distributed, use quadratic energy function

$$H(s^1;\Theta^1) := \frac{1}{2}(v^1 - b^{v^1})^\top (\Sigma^1)^{-1}(v^1 - b^{v^1}) - (v^1)^\top (\Sigma^1)^{-1/2} W^1 h^1 - (b^{h^1})^\top h^1$$

with diagonal covariance matrix Σ^1.

Appendix: Pre-Optimization

- For an output RBM, we use extra visual output units for pre-trainig h to have maximize asymptotically the discriminant criterion:
  - Outputs: $v^{out}(x) = W^{out} h(x) + b^{out}$
  - Targets: for $i = 1, \ldots, N$,

$$t_i^c := \begin{cases} \sqrt{N/N_c} & \text{if } y_i = c \\ 0 & \text{oterwise} \end{cases}$$

  - Minimizing MSE between $(v^{out}(x_i))_{i=1}^{N}$ and $(t_i)_{i=1}^{N}$ approximates the maximum of the discriminant criterion.

- Since the output RBM's visual output and hidden stated are modeled Gaussian-distributed, use an extended energy function

$$H(s;\Theta) := \frac{1}{2}(v^{out} - b^{out})^\top (\Sigma^{out})^{-1}(v^{out} - b^{out}) - (v^{out})^\top (\Sigma^{out})^{-1/2} W^{out} h + \frac{1}{2}(h - b^h)^\top (\Sigma^h)^{-1}(h - b^h) - (v^{in})^\top W^{in} (\Sigma^h)^{-1/2} h$$

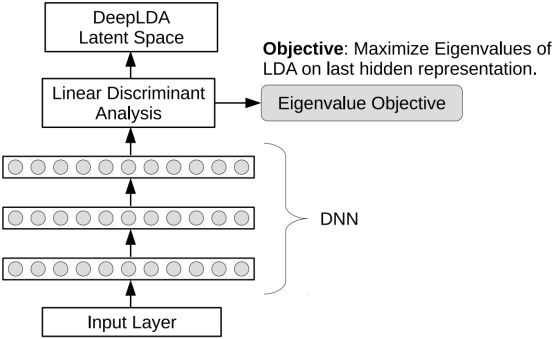
Deep Linear Discriminant Analysis

Dorfer, M., Kelz, R., & Widmer, G. (2015)

arXiv

Introduction

- Deep Linear Discriminant Analysis (DeepLDA) learns linearly separable latent representation in end-to-end fashion.
- Main idea: put LDA on top of a DNN to exploit the properties of classic LDA (low intra class variability, hight inter-class variablilty, optimal decision boundaries)



DeepLDA

- We want to produce features that show a low intra-class and high inter-class variability.
- Denote Θ as parameters of DNN and C is the number of classes.
- Objective functions:

$$\underset{\Theta}{\operatorname{argmax}} \frac{1}{C-1} \sum_{i=1}^{C-1} v_i$$

$\rightarrow$ It could be produce trivial solutions (maximize only the largest eigenvalue).

- DeepLDA's objective functions:

$$\underset{\Theta}{\operatorname{argmax}} \frac{1}{k} \sum_{i=1}^{k} v_i \text{ with } \{v_1, \ldots, v_k\} = \{v_j | v_j < \min\{v_1, \ldots, v_{C-1}\} + \epsilon\}, \quad (4)$$

where $\epsilon > 0$ is the margin.

Classification by DeepLDA

- X: training set, H: the topmost hidden representation on X
- A: LDA projection matrix
- $\bar{H}_c = (\bar{h}_1^\top, \ldots, \bar{h}_C^\top)$: per-class mean hidden representations
- For test sample x_t, compute h_t and define distances of h_t to the linear decision hyperplances:

$$d = h_t^\top T^\top - \frac{1}{2}\text{diag}(\bar{H}_c T^\top) \quad \text{with} \quad T = \bar{H}_c A A^\top,$$

where T are the decision hyperplane normal vectors.

- The vector of class probabilities for x_t:

$$p'_c = \frac{1}{1 + e^{-d}} \quad \rightarrow p_c = \frac{p'_c}{\sum p'_i}$$

Experimental Results

Method	Test Error
NIN + Dropout (Lin et al. (2013))	0.47%
Maxout (Goodfellow et al. (2013))	0.45%
DeepCNet(5,60) (Graham (2014))	0.31% (train set translation)
OurNetCCE(LDA)-50k	0.39%
OurNetCCE-50k	0.37%
OurNetCCE-60k	0.34%
DeepLDA-60k	0.32%
OurNetCCE(LDA)-60k	0.30%
DeepLDA-50k	**0.29**%
DeepLDA-50k(LinSVM)	**0.29**%

Figure : Comparison of test errors on MNIST

Max-Mahalanobis Linear Discriminant Analysis Networks

Tianyu Pang, Chao Du, Jun Zhu (2018)

Proceedings of the 35th International Conference on Machine Learning

Introduction

- For classification problems, DNNs with a softmax classifier are vulnerable to adversial attacks.
- Objective: design a robust classifier to adversarial attacks
- An adversarial example x^* crafted on x satisfies

$$\hat{y}(x^*) \neq \hat{y}(x), \quad \text{s.t.} \quad \|x^* - x\| \leq \epsilon,$$

where $\hat{y}(\cdot)$ denotes the predicted label from classifier, ϵ is the maximal perturbation.

Methodology

- Assumption 1: For the p-dimensional random vector x with its class label y,

$$P(y=i)=\pi_i, \quad P(x|y=i)=\mathcal{N}(\mu_i,\Sigma),$$

  where $i \in \{1,\ldots,C\}$, $\sum_i \pi_i = 1$ and each conditional Gaussian has the common Σ.

- Mahalanobis distance between any two Gaussian i and j defined as

$$\Delta_{i,j} = [(\mu_i - \mu_j)^\top \Sigma^{-1} (\mu_i - \mu_j)]^{\frac{1}{2}}$$

- W.L.O.G., assume Σ is nonsingular. Thus $\Sigma = QQ^\top$ where Q is a lower-triangular matrix.

Methodology

- Set $\tilde{x} = Q^{-1}(x - \bar{\mu})$ where $\bar{\mu} = \sum_i \mu_i / C$.
- Assumption 2: For the p-dimensional random vector x with its class label y,

$$P(y = i) = \pi_i, \quad P(\tilde{x}|y = i) = \mathcal{N}(\tilde{\mu}_i, I),$$

  where $i \in \{1, \ldots, C\}$, $\sum_i \pi_i = 1$ and $\sum_i \tilde{\mu}_i = 0$.
- Note that $\tilde{\Delta}_{i,j} = [(\tilde{\mu}_i - \tilde{\mu}_j)^\top (\tilde{\mu}_i - \tilde{\mu}_j)]^{\frac{1}{2}} = \Delta_{i,j}$.
- From now on, denote $x \leftarrow \tilde{x}$, $\mu_i \leftarrow \tilde{\mu}_i$ and $\Delta_{i,j} \leftarrow \tilde{\Delta}_{i,j}$.

Methodology

- Denote $\lambda_{i,j}(x) = 0$ as the decision boundary between class i and j obtained by LDA.
- Under the assumption 2, we randomly sample a normal example of class i as $x_{(i)}$ i.e., $x_{(i)} \sim \mathcal{N}(\mu_i, I)$, and denote its nearest adversarial as $x^*_{(i,j)}$ which is on the nearest decision boundary $\lambda_{i,j}(x) = 0$:

$$\hat{y}(x_{(i)}) = i, \quad \hat{y}(x^*_{(i)}) = j$$

- Define $d_{(i,j)} = d(x_{(i)}, x^*_{(i,j)})$.

Methodology

- **Theorem 1.** If $\pi_i = \pi_j$,

$$\mathbb{E}[d_{(i,j)}] = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\Delta_{i,j}^2}{8}\right) + \frac{1}{2}\Delta_{i,j}\left[1 - 2\Phi\Big(-\frac{\Delta_{i,j}}{2}\Big)\right],$$

where $\Phi(\cdot)$ is the normal c.d.f.

- Robustness of the classifier on all the attacks can be measured by

$$\mathrm{RB} = \min_{i,j \in \{1,\dots,C\}} \mathbb{E}[d_{(i,j)}]$$

- By Theorem 1, $|\mathbb{E}[d_{(i,j)}]/\Delta_{i,j} - 1/2|$ monotonically decreases to 0 w.r.t. $\Delta_{i,j}$, hence we can approximate RB as

$$\mathrm{RB} \approx \overline{\mathrm{RB}} = \min_{i,j} \Delta_{i,j}/2$$

Methodology

- **Theorem 2.** Assume that $\sum_{i=1}^{C} \mu_i = 0$ and $\max_i \|\mu_i\|_2^2 = L$. Then we have

$$\overline{\text{RB}} \leq \sqrt{\frac{LC}{2(C-1)}}.$$

The equality holds iff

$$\mu_i^\top \mu_j = \begin{cases} L, & i = j \\ L/(1-C), & \text{otherwise}, \end{cases} \qquad (5)$$

where $i, j \in \{1, \dots, C\}$.

Methodology

- Denote μ^* as any set of means that satisfy the optimal condition (5).
- With the previous results, LDA classifier have the best robustness if its input distribution is

$$P(y = i) = \pi_i, \quad P(x|y = i) = \mathcal{N}(\mu_i^*, I), \quad i = 1, \dots, C$$

- But, in general, the mixture of Gaussian assumption does not hold in the input space.

Max-Mahalanobis LDA Networks

- By exploring the power for DNNs, we propose the **Max-Mahalanobis linear discriminant analysis (MM-LDA)** network, which consists of
  - a nonlinear transformation network $x \mapsto z_\theta$ parametrized by θ;
  - applied the MM-LDA procedure on z_θ.
- Given a feautre vector z_θ, the conditional distribution of labels is

$$P(y = k | z_\theta) = \frac{\pi_k \mathcal{N}(z_\theta | \mu_k^*, I)}{\sum_{i=1}^{L} \pi_i \mathcal{N}(z_\theta | \mu_i^*, I)}.$$

- Finally, θ are trained by using cross-entropy loss function.

Max-Mahalanobis LDA Networks

Algorithm 2 The training phase for the MM-LDA network

Input: The model $z_\theta(x)$, the square norm C of Gaussian means, the training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i\in[N]}$.

Initialization: Initialize θ as θ_0, the training step as $s = 0$. Let $p = \dim(z)$, ε be the learning rate variable.

Get $\mu^* = \text{GenerateOptMeans}(C, p, L)$ for the MMD.

while not converged **do**

Sample a mini-batch of training data $\mathcal{D}_m$ from $\mathcal{D}$,

Calculate the objective

$$\mathcal{L}_{\text{MM}}^m = \frac{1}{|\mathcal{D}_m|} \sum_{(x_i, y_i)\in\mathcal{D}_m} \mathcal{L}_{\text{MM}}(x_i, y_i, \mu^*),$$

Update parameters $\theta_{s+1} \leftarrow \theta_s - \varepsilon\nabla_\theta \mathcal{L}_{\text{MM}}^m$,

Set $s \leftarrow s + 1$.

end while

Return: The parameters $\theta_{\text{MM}} = \theta_s$.

Experiment Results

Perturbation	Model	MNIST FGSM	MNIST BIM	MNIST ILCM	MNIST JSMA	CIFAR-10 FGSM	CIFAR-10 BIM	CIFAR-10 ILCM	CIFAR-10 JSMA
0.04	Resnet-32 (SR)	93.6	87.9	94.8	92.9	20.0	5.5	0.2	65.6
	Resnet-32 (SR) + SAT	86.7	68.5	98.4	-	24.4	7.0	0.4	-
	Resnet-32 (SR) + HAT	88.7	96.3	**99.8**	-	30.3	5.3	1.3	-
	Resnet-32 (MM-LDA)	**99.2**	**99.2**	99.0	**99.1**	**91.3**	**91.2**	**70.0**	**91.2**
0.12	Resnet-32 (SR)	28.1	3.4	20.9	56.0	10.2	4.1	0.3	20.5
	Resnet-32 (SR) + SAT	40.5	8.7	88.8	-	88.2	6.9	0.1	-
	Resnet-32 (SR) + HAT	40.3	40.1	92.6	-	44.1	8.7	0.0	-
	Resnet-32 (MM-LDA)	**99.3**	**98.6**	**99.6**	**99.7**	**90.7**	**90.1**	**42.5**	**91.1**
0.20	Resnet-32 (SR)	15.5	0.3	1.7	25.6	10.7	4.2	0.6	11.5
	Resnet-32 (SR) + SAT	17.3	1.1	69.4	-	**91.7**	9.4	0.0	-
	Resnet-32 (SR) + HAT	10.1	10.5	46.1	-	40.7	6.0	0.2	-
	Resnet-32 (MM-LDA)	**97.5**	**97.3**	**96.6**	**99.6**	89.5	**89.7**	**31.2**	**91.8**