PCA+KNN and Normalized SL on Modified MNIST data

Geetartho Chanda Anindita Deb Daksh Khorana Soyun Park

Abstract

The goal of the project is to improve classification performance on the modified MNIST data with 10 digits (classes). By changing the number of images in each class (imbalanced dataset) or adding noise via a class exchange rule (symmetric or asymmetric noisy dataset), we compare the proposed methods based on PCA+KNN and Normalized SL with the baseline methods such as support vector machine, logistic regression, LDAM-DRW and SL. We show improvements in classification based on performance measures, e.g., accuracy, precision, recall, f1 score and AUC.

1. Introduction

We have a popular MNIST dataset for digit classification using Machine Learning (ML) and Deep Learning (ML) techniques. Examples of ML techniques are categorized depending on whether data have a response variable (y) or multiple response variables (y's). When a response variable is not available, the learning is called unsupervised learning whereas, if a response variable is used, it is called supervised learning. Unsupervised learning is mostly aimed for exploratory data analysis to discuss the characteristics of covariates (x's). For example, clustering is one of the frequently used unsupervised learning method. On the other hand, supervised learning focuses on explaining direct or indirect (latent) relationships between y's and x's, which is regression and classification. While regression shows how covariates explain or predict a continuous response variable, classification address the same procedure for a categorical or binary response variable. In the MNIST dataset, there are 10 digit from 0 through 9 as categorical response variables and images that is comprised of 28×28 pixels as 784 covariates. The goal of the study is to find the best classification technique as possible by training the model based on logistic regression, support vector machines (SVMs), convolutional neural networks (CNNs), and their extensions, (e.g., LDAM-DRW [1] for regularizing unbalanced classes and Symmetric Learning (SL) [10] for adjusting to noisy data) and classifying the handwritten digits. The process can be implemented with three goals in mind. First, we improve the ability of classification based on the performance measures - accuracy, recall, precision, f1 and AUC. See Section 3.1. Second, we develop flexible methods that are adaptable to imbalanced dataset, which has unequal numbers of categories in the response variables, and to noisy dataset whose rule for assigning the digit is changed with some randomness. Third, we reduce the computation load –for example, computation time, training loss, and test loss. We propose a DL model based on "Normalized active-passive loss function". The baseline methods to be compared with proposed methods are logistic regression, SVM, LDAM-DRW and SL.

2. Methods

2.1. Literature Review

We reviewed literature related to ML/DL techniques. LeCun *et al*. (1998) [5] introduced LeNet architecture and compare it with other machine learning techniques. Cao *et al*. (2019) developed LDAM-DRW to address imbalanced data [1].

The paper [1] proposes methods to learn imbalanced data sets with label distribution aware margin loss. It proposes that we replace the loss with a cross-entropy objective during training, which is feasible with standard training strategies. As we have less data on minority classes and the current models are enormous and overfitting to minorities, the idea is to regularize the minority classes so that the frequency of the class can be improved. To implement this, we need a data-dependent or label-dependent regularizer that can be obtained using a large margin for minority edges and minimum margin per class and uniform label test error bounds. Cost-sensitive re-weighting and re-sampling are two well-known and successful strategies to cope with imbalanced datasets because, in expectation, they effectively make the imbalanced training distribution closer to the uniform test distribution. The main algorithm proposed in this paper is LDAM-DRW. Which can be used in conjecture to optimize consistent label generalization error bound.

Wang *et al*. (2019) [10] developed SL algorithms for noisy data. The paper mentions that learning using Cross Entropy (CE) loss on noisy data leads to overfitting on

some ('easy') classes and significant underfitting on some other ('hard') classes. The paper proposes the introduction of noise intolerant Reverse Cross Entropy (RCE) term to reduce overfitting of easy classes and underfitting of hard classes and applies it to Cross Entropy to obtain Symmetric Cross Entropy. For further robustness of learning, two decoupled hyper-parameters (α, β) are added, to address overfitting and to improve the flexibility of RCE's robustness.

SL facilitates adaptive learning rate to improve learning for hard classes and decrease learning rate for classes with probability greater than 0.5 and increase learning rate for classes with probability less than 0.5.

2.2. Data Explanation

We use the MNIST dataset which comes with the default training set of images having 60,000 images and the test size of 10,000 images. Each image is of dimension 28×28. The data is classified into 10 digit classes having numbers [0,1,2,3,4,5,6,7,8,9]. The original dataset is modified by balancing it and adding symmetric/asymmetric noise. Balancing is done using randomly oversampling samples of classes with lower representation to match that of the class with the most number of samples. Symmetric noisy labels are generated by flipping the labels of a given proportion of training samples to one of the other class labels uniformly. Whilst for asymmetric noisy labels, flipping labels only occurs within a specific set of classes, for example, for MNIST, flipping $2 \rightarrow 7, 3 \rightarrow 8, 5 \leftrightarrow 6$ and $7 \rightarrow 1$ [10].
We use 20% of the training data for validation while training and leave the test dataset untouched.

2.3. Baseline Methods

2.3.1 Support Vector Machine (SVM)

- SVM is generally used for classification problems but it can be used for classification as well as regression. In SVM each data item is plotted in a n-dimensional(n is feature) space having each feature as a coordinate, and the classification is performed by finding hyper-plane that differentiates the classes [3].

$$\min_{w,b} \frac{1}{2}||w||^2 \text{ subject to } y_i(w \cdot x + b) - 1 \geq 0, i = 1, ..., m \tag{1}$$

2.3.2 Logistic Regression

- Logistic Regression or LR is one of the basic supervised learning model to be used in Machine Learning. The classification done by LR gives out binary output i.e 0 or 1, a sigmoidal function **??** is used to get the result. The general equation for Logistic Regression is given as below.

$$\begin{aligned} L(\beta; \mathbf{y}, \mathbf{X}) &= \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i} \\ &= \prod_{i=1}^{n} \left(\frac{\exp(\mathbf{X}_i \beta)}{1 + \exp(\mathbf{X}_i \beta)} \right)^{y_i} \left(\frac{1}{1 + \exp(\mathbf{X}_i \beta)} \right)^{1 - y_i}. \end{aligned} \tag{2}$$

2.3.3 LDAM-DRW [1]

- The main focus of this paper is on imbalanced datasets especially when the training dataset is heavily imbalanced *i.e.*, some classes have more objects than others. For improved performance of DL models in such scenarios, an effective way for generalization on less frequent class objects is required. The paper achieves this feat by enforcing a larger margin γ_j to the classes with lower number of samples n_j.

- It introduces the label-distribution-aware margin (LDAM) loss to minimize a margin-based generalization bound, which replaces the default cross-entropy loss during training. This loss function is applied to the model along with a prior strategy of re-weighting. A deferred re-weighting (DRW) strategy is applied during training to defer re-balancing of the weights until after the initial stage thus, letting the model pick up on the initial representation of the model before the weights are re-balanced.

- For two classes, the class distribution-aware margins derived by the authors is given by

$$\gamma_1 = \frac{C}{n_1^{1/4}} \text{ and } \gamma_2 = \frac{C}{n_2^{1/4}}, \tag{3}$$

and its multi-class extension is given by

$$\gamma_j = \frac{C}{n_j^{1/4}}, \tag{4}$$

where C is a constant and a hyper-parameter that can be tuned. The LDAM loss is implemented using the cross entropy loss with enforced margins defined by

$$\mathcal{L}_{LDAM}((x, y); f) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \Sigma_{j \neq y} e^{z_j}}, \tag{5}$$

where $\Delta_j = \frac{C}{n_j^{1/4}}$ for $j \in \{1, ..., k\}$.

2.3.4 SL [10]

- This method focuses on improving the performance of deep neural networks in the presence of noisy labels DNN learning with Cross Entropy loss performs

poorly even with clean labels as it overfits to some (easy) classes whereas for some other (hard) classes it converges much slower. In the presence of noise, this phenomenon is amplified significantly. The paper achieves better learning by adding an extra, noise-robust term to the CE loss function.

- The basis of symmetric learning, inspired by symmetric KL-divergence, is the addition of the Reverse Cross Entropy term to the Cross Entropy loss. This simultaneously addresses the under learning and overfitting problems faced by CE.

$$\ell_{sl} = \alpha \times \ell_{ce} + \beta \times \ell_{rce}, \tag{6}$$

where α addresses overfitting and β improves flexibility of RCE's robustness.

2.4. Proposed Methods

2.4.1 Proposed ML Method: PCA+KNN

- Why did we choose PCA with KNN?
  - KNN does not work well with large dataset.
    In large datasets, the cost of calculating the distance between the new point and each existing point is huge which degrades the performance of the algorithm.
  - KNN does not work well with high dimensions.
    The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension. It is sensitive to high level of noise. It does not eliminate noise, but it can reduce noise.
- KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g., SVM, Linear Regression etc. Also, KNN is very easy to implement. There are only two parameters required to implement KNN that is the value of K and the distance function (e.g., Euclidean distance).
- A PCA-KNN model is composed of two technical components.
  PCA for reducing the data dimensionality and redundancy information while remaining principal components with rich-information, and KNN for MNIST dataset. Especially, PCA transforms the input data to a set of principal components as input for prediction that can reduce the calculation and improve the performance of KNN. We transform the original feature space to an n dimensional space which is orthogonal to the original space.
- Let $X \in R_{n\times m}$ denote the raw data matrix with n samples (rows) and m variables (columns). X is first scaled to zero mean for covariance-based PCA and further to unit variance for correlation-based PCA. By a singular value decomposition (SVD) algorithm, the scaled matrix X is decomposed for the response variable Y and covariates X as follows.

$$Y = ZU = (U\Sigma V^T)^T U = V\Sigma^T, \tag{7}$$

$$X = TP^T + \bar{X} = TP^T + \bar{T}\bar{P}^T = (T\bar{T})(P\bar{P})^T, \tag{8}$$

where $T \in \mathbb{R}^{n\times l}$ and $P \in \mathbb{R}^{m\times l}$ are the score and loading matrices, respectively. The PCA projection reduces the original set of m variables to l principal components (PC's) [4].

2.4.2 Proposed DL Method: Normalized SL

For our proposed DNN model we use the Active Passive Loss framework (APL) as defined by [7] which combines the loss functions that act as a catalyst for each other. The paper states that a simple normalization of loss functions make them more robust to noisy labels. We implement this idea by using the properties of Symmetric Learning [10] loss where Cross Entropy (Active loss) and Reverse Cross Entropy (Passive Loss) are combined but instead, we normalize the terms of the loss function. The normalized loss function proposed by [7] is

$$\mathcal{L}_{norm} = \frac{\mathcal{L}(f(x), y)}{\Sigma_{j=1}^{K}\mathcal{L}(f(x), j)}. \tag{9}$$

Following this scheme, our proposed loss function is

$$\mathcal{L}_{SLnorm} = \alpha \times \ell_{CE} + \beta \times \ell_{RCEnorm}. \tag{10}$$

We implement a 3 layer CNN model followed by a fully connected layer for the original as well as the symmetric/asymmetric noisy and balanced/unbalanced MNIST dataset [1, 6, 10] and compare our performance with the models previously discussed. Parameters for the baseline models are configured as per their papers. For our model we set $\alpha = 0.1, \beta = 1$ and $A = -4$ [10]. Training is done using SGD with momentum 0.9, weight decay 10^{-4} and initial learning rate 0.1. Learning rate is divided by 10 after 10 and then again after 30 epochs. Total number of epochs is 150 for the original balanced and imbalanced dataset and 200 for the remaining four dataset variations. The training is done in batches of 128 for 40% noisy dataset [7].

3. Experiments and Results

3.1. Prediction Results

We report performance measures based on accuracy (ACC), precision (PC), recall (RC) (= sensitivity (SEN)), specificity (SPF) and Area Under the Curve (AUC). According to the traditional confusion matrix, TP, FP, TN and FN refer to True Positive, False Positive, True Negative and False Negative, respectively. The equations are showing the related formulas below.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$PC = \frac{TP}{TP + FP} \tag{12}$$

$$RC = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = \frac{2 \times PC \times RC}{PC + RC} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{14}$$

$$SPF = \frac{TN}{FP + TN} \tag{15}$$

Lastly, AUC is calculated based on 13 and 15, which is the area under the receiver operating characteristic (ROC) curve comprising of SEN in a x axis and $1 - SPF$ in a y axis. The results from the baseline methods and proposed method are shown below. The tables 1-5 show ACC, PC, RC, f1 score and AUC from the baseline methods (SVM, LR, LDAM and SL) and proposed methods (normalized symmetric noise and KNN+PCA) fit on six different datasets produced by a combination of Balanced (B), Imbalanced (I), Symmetric (S) and Asymmetric (A). The table 6 shows when $k = 45$ and 35 components by the proposed ML model on asymmetric data, while tables 1-5 shows when $k = 9$ and 35 components for the proposed ML model.

On comparing SVM and Logistic Regression (LR) models are for all the six datasets, it is observed that SVM is giving a high performance compared to the LR models. However both these models are sensitive to noisy datasets. On implementation of LDAM-DRW, we can see that SVM and LR become less sensitive to imbalanced dataset and give out robust performance that is comparable to the original imbalanced dataset. Similarly, on implementation of CrossEntropy+ReverseCrossEntropy loss functions, we observe that SVM and LR are less sensitive to noisy dataset and provides robust performance similar to original dataset with noise rate 0%. However, it is empirically observed that for the symmetric noisy dataset and the asymmetric noisy dataset. DL models performed better in the former case in terms of ACC, PC, RC and F1 compared to that for the symmetric noisy dataset.

In addition to the above observations, we found that though these robust functions are reflecting stable and consistent performance on all the six types of datasets but they suffer from a problem of under fitting. To address this, we propose a framework to build robust loss functions called Active Passive Normalized Loss.

The Proposed DL model uses normalized implementation of SL loss that shows good results on the noisy data. The Active Passive framework which is used in our DL shows that it has a very good result when compared to state of the art methods on benchmark datasets, this result is empirically verified. The Active Passive Framework can serve as a benchmark to develop new robust loss function.

The proposed ML part uses KNN along with PCA, that gives out moderate accuracy on our dataset having 784 dimensions with reduced dimensionality. It is empirically verified that using PCA along with KNN does not reduce noise level rather we chose high level of K to achieve good performance on noisy dataset.

After applying PCA, we are getting a maximum variance of 90% explained by 200 features for all the datasets. Using the 200 principal components, we fitted the KNN Model. However, we observed that using just 35 principal components to fit the KNN model we can achieve the same accuracy level as that with 200 principal components and hence we further reduced the number of features to fit KNN model. From the results, its observed that for Imbalanced Asymmetric and Balanced Asymmetric the accuracy is within range $80 - 82\%$ for $k = 11$ and the number of features, 35, and which is quite low compared to other datasets for symmetric noise its comparable to that for original balanced and imbalanced dataset with $k = 9$ and the number of components, 35. On increasing k to 45 for imbalanced asymmetric dataset, we are getting an accuracy of 87%.Thus for higher values of k we can still get an improved accuracy for noisy dataset. This recommendation was mentioned in the the paper [9].

Data Type	B	I	BS	BA	IS	IA
SVM	0.9257	0.9239	0.9588	0.8803	0.9567	0.8637
LR	0.9193	0.9201	0.8807	0.8129	0.8748	0.8122
LDAM-DRW	0.8553	0.9526	0.6941	0.8757	0.7137	0.8553
SL	0.9933	0.9933	0.9717	0.9754	0.9781	0.9770
Proposed ML	0.9504	0.949	0.8859	0.8176	0.9264	0.8190
Proposed DL	0.9718	0.9751	0.9707	0.9463	0.9613	0.9588

Table:1 Accuracy 11

Data Type	B	I	BS	BA	IS	IA
SVM	0.9600	0.9600	0.9800	0.9400	0.9800	0.9300
LR	0.9500	0.9500	0.9200	0.8500	0.9200	0.8500
LDAM-DRW	0.9894	0.9403	0.8100	0.9109	0.9000	0.9894
SL	0.9933	0.9933	0.9740	0.9782	0.9805	0.9797
Proposed ML	0.9800	0.9800	0.9300	0.7700	0.9200	0.7800
Proposed DL	0.9722	0.9758	0.9718	0.9471	0.9633	0.9594

Table:2 Precision 12

Data Type	B	I	BS	BA	IS	IA
SVM	0.9257	0.9239	0.9588	0.8803	0.9567	0.8637
LR	0.9193	0.9201	0.8807	0.8129	0.8748	0.8122
LDAM-DRW	0.1771	0.0230	0.6500	0.0778	0.7100	0.1771
SL	0.9931	0.9931	0.9687	0.9726	0.9766	0.9739
Proposed ML	0.9900	0.9900	0.9800	0.9900	0.9900	0.9900
Proposed DL	0.9715	0.9749	0.9699	0.9456	0.9606	0.9576

Table:3 Recall 13

Data Type	B	I	BS	BA	IS	IA
SVM	0.9257	0.9239	0.9600	0.8800	0.9600	0.8600
LR	0.9193	0.9201	0.8800	0.8100	0.8700	0.8100
LDAM-DRW	0.2953	0.7617	0.6300	0.1412	0.7600	0.2953
SL	0.9803	0.9651	0.9718	0.8448	0.9433	0.8277
Proposed ML	0.9800	0.9800	0.9500	0.8600	0.9500	0.8700
Proposed DL	0.9718	0.9754	0.9708	0.9463	0.9619	0.9585

Table:4 F1 Score 14

Data Type	B	I	BS	BA	IS	IA
SVM	0.9892	0.9888	1.000	0.9900	1.0000	0.9900
LR	0.9886	0.9887	0.9800	0.9700	0.9800	0.9700
LDAM-DRW	0.9900	0.9900	0.8312	0.9895	0.8400	0.9900
SL	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900
Proposed ML	0.9700	0.9600	0.9300	0.9200	0.9400	0.9900
Proposed DL	0.9730	0.9836	0.9664	0.9693	0.9611	0.9723

Table:5 AUC 3.1

Performance Measure	PC	RC	F1	AUC
BA	0.8600	0.9900	0.9200	0.9200
IM	0.8400	0.8400	0.9300	0.9300

Table:6 k=45 and 35 Components on Asymmetric Data

4. Conclusion

The major advantage we have in this model is that it can deal with real unprecedented data and most of the time the given labels are imbalanced. The noise in the data also affects the quality of the data and makes it harder for the Machine Learning and Deep Learning performance while training as well as predicting the data.

The disadvantages in this are parameter tuning as due to the noisy labels we are unsure of the data which makes it harder to decide that which parameters are to be changed and modified, and so we have to tune them manually and

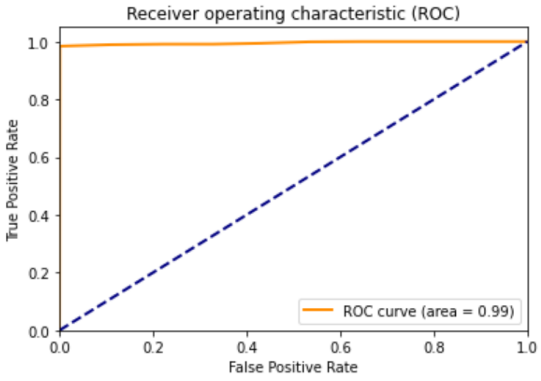

Figure 1. ROC Curve from SL on Balanced data

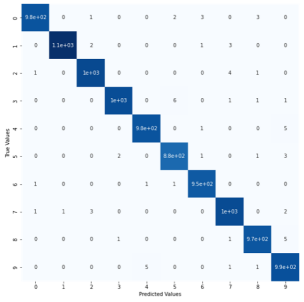

Figure 2. Confusion Matrix from SL on Balanced data

test for each output if it is feasible or not which makes it time consuming. The optimal nodes and layers in this case are never known or can be predicted before hand.

While the current technology provides a high resolution in pictures and videos, e.g., $3,840 \times 2,160$ pixels (4k), $1,920 \times 1,080$ pixels (Full HD) or even $7,680 \times 4,320$ (8k), our dataset has only 28×28 pixels that might not appear in real life. The sample size of 60,000 is enormously large although it is very hard and expensive to gather large sample imaging data in the research fields such as clinical studies. We might cast a doubt on the applicability of the methods and the performance using the existing methods. Using extremely clean dataset, the test accuracy over 99 and the AUC over 99 might be possible, but there is not much significance difference between other approaches. With any of those methods we used in the paper, it might be hard to distinguish them from perspectives of performance measures. We have to find realistic ML/DL techniques [2] that can be applied to real data with a much less sample size, for example, n=92 in PET imaging data [8].

5. Contributions

We thank everyone in the team for their wonderful contributions. Anindita Deb and Geetartho Chanda contributed to setting up Python codes (baseline methods and proposed

methods) and writing up methods/conclusion parts. Soyun Park and Daksh Khorana contributed to implementing the codes, organizing the results, formatting, editing the documents and searching literature.

References

[1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019. 1, 2, 3

[2] Richard E Carson. Tracer kinetic modeling in pet. In *Positron Emission Tomography*, pages 127–159. Springer, 2005. 5

[3] Shuzhan Fan. SVM formulation, 2018. 2

[4] Q. Peter He and Jin Wang. Principal component based k-nearest-neighbor rule for semiconductor process fault detection. In *2008 American Control Conference*, pages 1606–1611, 2008. 3

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[6] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 3

[7] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020. 3

[8] Evan D Morris, Christopher J Endres, Kathleen C Schmidt, Bradley T Christian, Raymond F Muzic, and Ronald E Fisher. Kinetic modeling in positron emission tomography. *Emission tomography*, 46:499–540, 2004. 5

[9] Stefanos Ougiaroglou and Georgios Evangelidis. Dealing with noisy data in the context of k-nn classification. In *Proceedings of the 7th Balkan Conference on informatics conference*, BCI '15, pages 1–4. ACM, 2015. 4

[10] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 1, 2, 3