Final Project Report – 236802

Abstract

Deep neural networks (DNNs) have achieved substantial success in a variety of applications across many disciplines (like consumers, medical, etc.). Yet, their great performance comes with the expensive cost of requiring correctly annotated large-scale datasets, which is one of major obstacles of using them. Moreover, due to DNNs' rich capacity, errors in training labels can hamper performance. Manual labeling or reannotation becomes impractical, when large amount of data is used, and therefore creative approach is required. Nam et. al.[1] suggest a creative twofold approach of intentionally train the first network to be biased by repeatedly amplifying its "prejudice", and simultaneously debias the training of the second network by focusing on samples that go against the prejudice of the first biased network. My research further expands this approach by assessing the performance of the loss function which amplifies the "prejudice" (generalized cross-entropy) and assessing this approach on physiological database, which is highly prone to bias.

Section 1: Background and problem setup

The resurrection of neural networks in recent years, together with the recent emergence of large-scale datasets, has enabled super-human performance on many classification tasks. However, neural networks are not free of caveats. Among other difficulties in training neural networks, one of the main difficulties is that neural networks, especially deep ones, often learn to make predictions that overly rely on spurious correlation existing in the dataset, which causes the model to be biased. Numerous methods have been proposed for learning with noisy labels with DNNs in recent years, and particularly dealing with bias. One of the prevalent approaches shown in the previous work to tackle this issue is explicitly labeling of misleadingly correlated attributes.[2] This attributes might be given during training as side information.[3] The other approach would be presuming a particular bias type. The latter approach, however, requires characteristics and diagnostics by human experts.[4] Therefore, these two approaches aren't cost-effective, and sometimes even impractical.

Section 2: The chosen paper

The paper I chose to focus on named " Learning from Failure: Training Debiased Classifier from Biased Classifier" of Nam et. al. suggests a very interesting, yet cost effective and practical way of overcoming bias[1]. Briefly, the idea relies on two main assumptions:

(1) Neural networks learn to rely on the spurious correlation only when it is "easier" to learn than the desired knowledge.
(2) Such reliance is most prominent during the early phase of training.

Given these assumptions, Nam et. al. suggested a twofold approach:

(a) Train the first network to be biased by repeatedly amplifying its "prejudice" (f_B)
(b) Debias the training of the second network by focusing on samples that go against the prejudice of the biased network in (a)- (f_D).

Schematically, this architecture can be depicted as shown in figure 1.

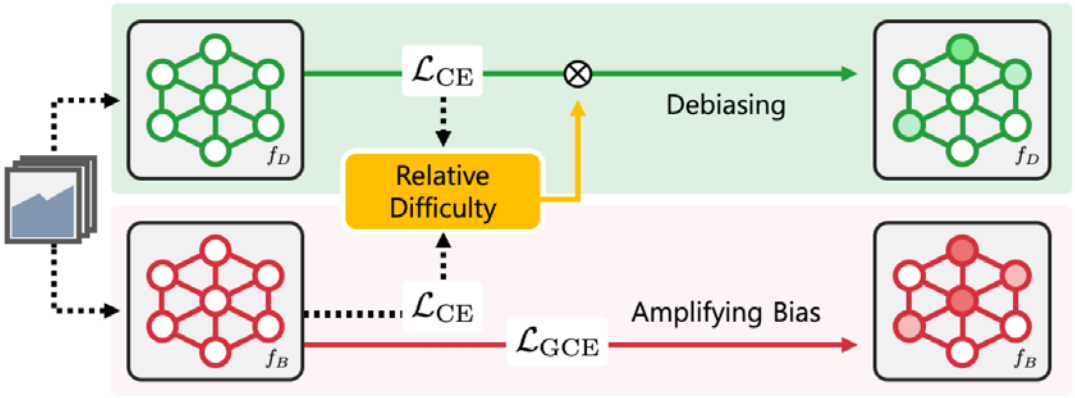

Figure 1: Illustration of training two models (f_D; f_B) to be debiased and biased, respectively. The biased model optimizes generalized cross entropy (L_{GCE}) loss to amplify bias. The debiased model trains with weighted cross entropy loss leveraging relative difficulty. It results in larger weights to bias-conflicting samples while training the debiased model.

The "bias" amplification in biased network (f_B) is done by replacing cross entropy loss function by generalized cross entropy loss.

Generalized cross entropy is defined as following:

$$\mathrm{GCE}(p(x;\theta),y) = \frac{1-p_y(x;\theta)^q}{q}$$

where $p(x;\theta)$ and $p_y(x;\theta)$ are softmax output of the neural network and its probability assigned to the target attribute of y, respectively. Here, $q \in (0,1]$ is a hyperparameter that controls the degree of amplification. Compared to the CE loss, the gradient of the GCE loss up-weights the gradient of the CE loss for the samples with a high probability p_y of predicting the correct target attribute as follows:

$$\frac{\partial \mathrm{GCE}(p,y)}{\partial \theta} = p_y^q \frac{\partial \mathrm{CE}(p,y)}{\partial \theta}$$

The cross-entropy generalization was broadly discussed at Zhang et. al.[5] , including a detailed analysis on ResNet deep neural network, and a conclusion that GCE sample equally, which makes it more robust to noisy labels. However, as they demonstrated empirically **(and I will prove once again in results section**), this can lead to significantly longer training time before convergence. Moreover, without the implicit weighting scheme to focus on challenging samples, the stochasticity involved in the training process can make learning difficult. Deep discussion of generalized cross entropy is beyond this final project summary, but on a short notice when $q \to 0$, using L'Hôpital's rule, the proposed generalized cross entropy loss function is equivalent to cross entropy loss.

While certain batch forward-propagates through biased network f_B, the paper suggests training a debiased model simultaneously with the samples using the CE loss re-weighted by the following relative difficulty score:

$$\mathcal{W}(x) = \frac{\mathrm{CE}(f_B(x), y)}{\mathrm{CE}(f_B(x), y) + \mathrm{CE}(f_D(x), y)}$$

where $f_B(x)$; $f_D(x)$ are softmax outputs of the biased and debiased model, respectively.

Briefly, the algorithm of the paper I chose can be summarized by the following steps:

1: **Input**: θ_B, θ_D, training set $\mathcal{D}$, learning rate η, number of iterations T
2: Initialize two networks $f_B(x; \theta_B)$ and $f_D(x; \theta_D)$.
3: **for** $t = 1, \cdots, T$ **do**
4: Draw a mini-batch $\mathcal{B} = \{(x^{(b)}, y^{(b)})\}_{b=1}^{B}$ from $\mathcal{D}$
5: Update $f_B(x; \theta_B)$ by $\theta_B \leftarrow \theta_B - \eta \nabla_{\theta_B} \sum_{(x,y)\in\mathcal{B}} \mathrm{GCE}(f_B(x), y)$.
6: Update $f_D(x; \theta_D)$ by $\theta_D \leftarrow \theta_D - \eta \nabla_{\theta_D} \sum_{(x,y)\in\mathcal{B}} \mathcal{W}(x) \cdot \mathrm{CE}(f_D(x), y)$.
7: **end for**

Section 3: Creative extension

Main goal of the extension is twofold:

(1) To assess the expected behavior described in of generalized cross entropy loss on physiological dataset (ECG), as described in the prior-art of the suggested paper[5], i.e. reproducing the results of Zhang et. al. on physiological database.

(2) To empirically verify the suggested bias elimination approach on physiological data (ECG).

The reason is that ECG data labeled by humans will most likely comprise bias due to native challenge of accurately classifying more than 100 diseases, and built-in similarity between various cardiac disorders. For this purpose, I chose to use the NYU School of Medicine, Emergency Care Electrocardiogram (ECG) Database which comprises more than 80000 records of more than 100 diseases (not with the same prevalence though). A typical record looks like shown in fig. 2. More details about the database can be found on database site.[6]

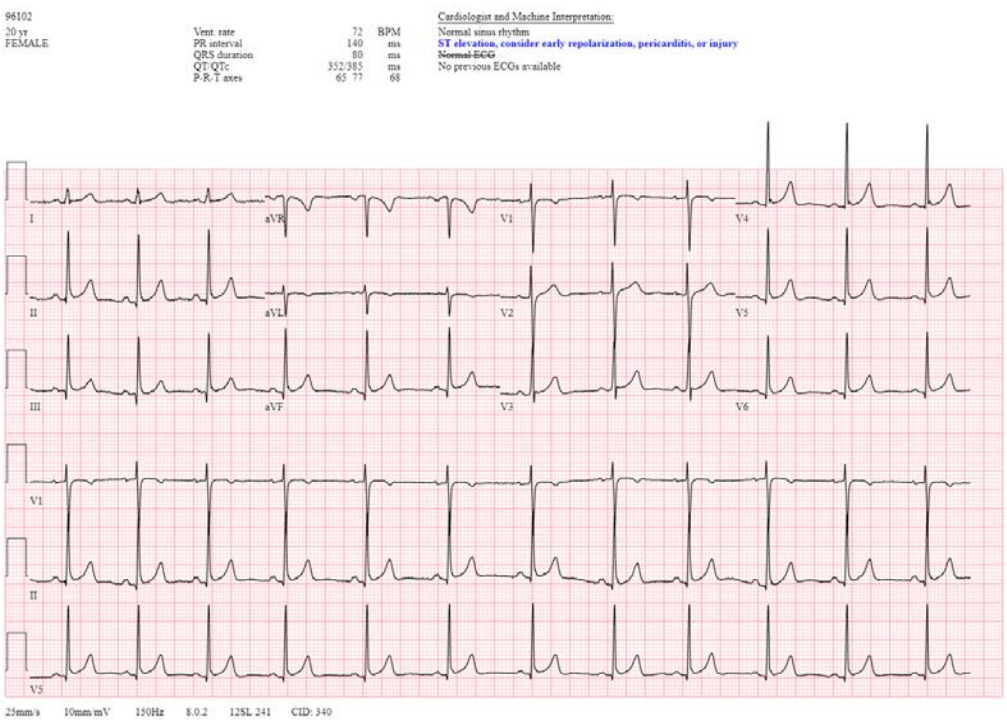

Figure 2: Typical 12 lead ECG plot image from NYU School of Medicine, Emergency Care Electrocardiogram (ECG) Database. At the middle-top section you can see the

Since not all diseases exist in the database with the same prevalence, I made a histogram of the disease tags and depicted 20 most prevalent categories of the database in fig. 3. Being noticed that 2 most prevalent categories aren't cardiac diseases, but rather healthy condition description (sinus rhythm and normal ECG). Therefore, in my analysis I will refer to the next 18 most prevalent diseases.

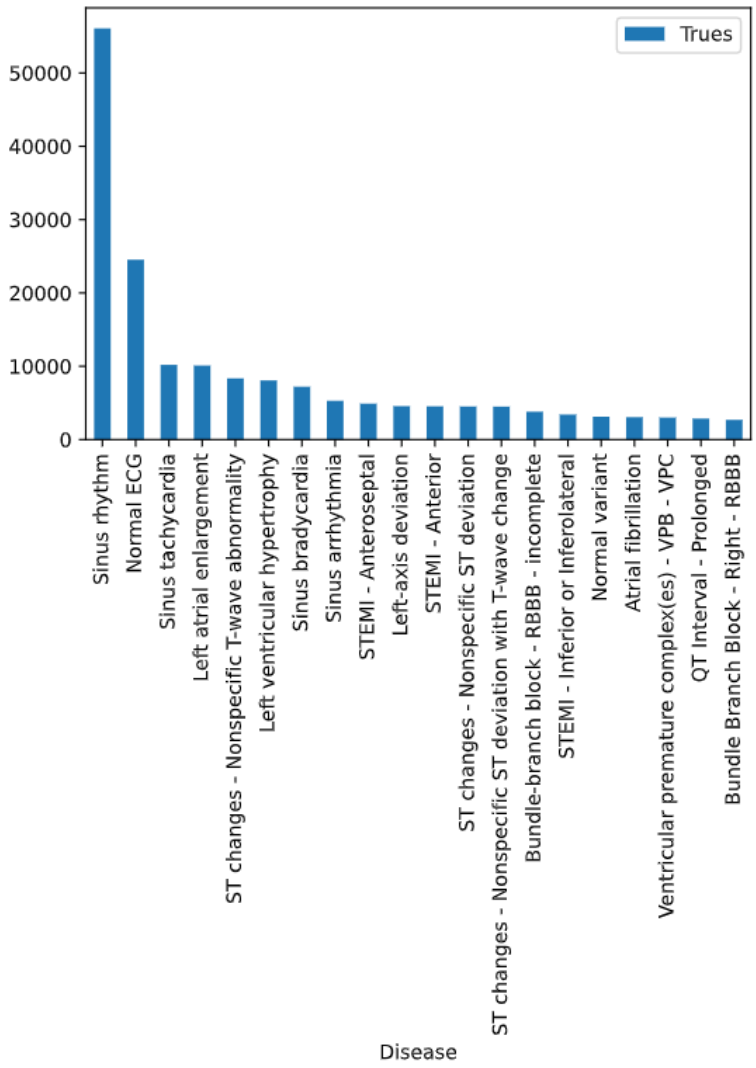

Figure 3: Most prevalent disease categories in the database. First two categories (Sinus rhythm and Normal ECG) are not a disease names, but healthy conditions. The next 18 are disease names. Source code can be found in "Main_notebook.ipynb" in project code repository.[7]

To find categories which are bias suspicious, I built a correlogram of most 18 prevalent diseases (as mentioned before, "sinus rhythm" and "normal ECG" aren't diseases). The correlogram results are shown in fig.4. It turns out that there is a suspicious correlation between "Left ventricular hypertrophy" and "Normal variant". The correlation is 0.52, which is extraordinary in comparison to all other category permutations.

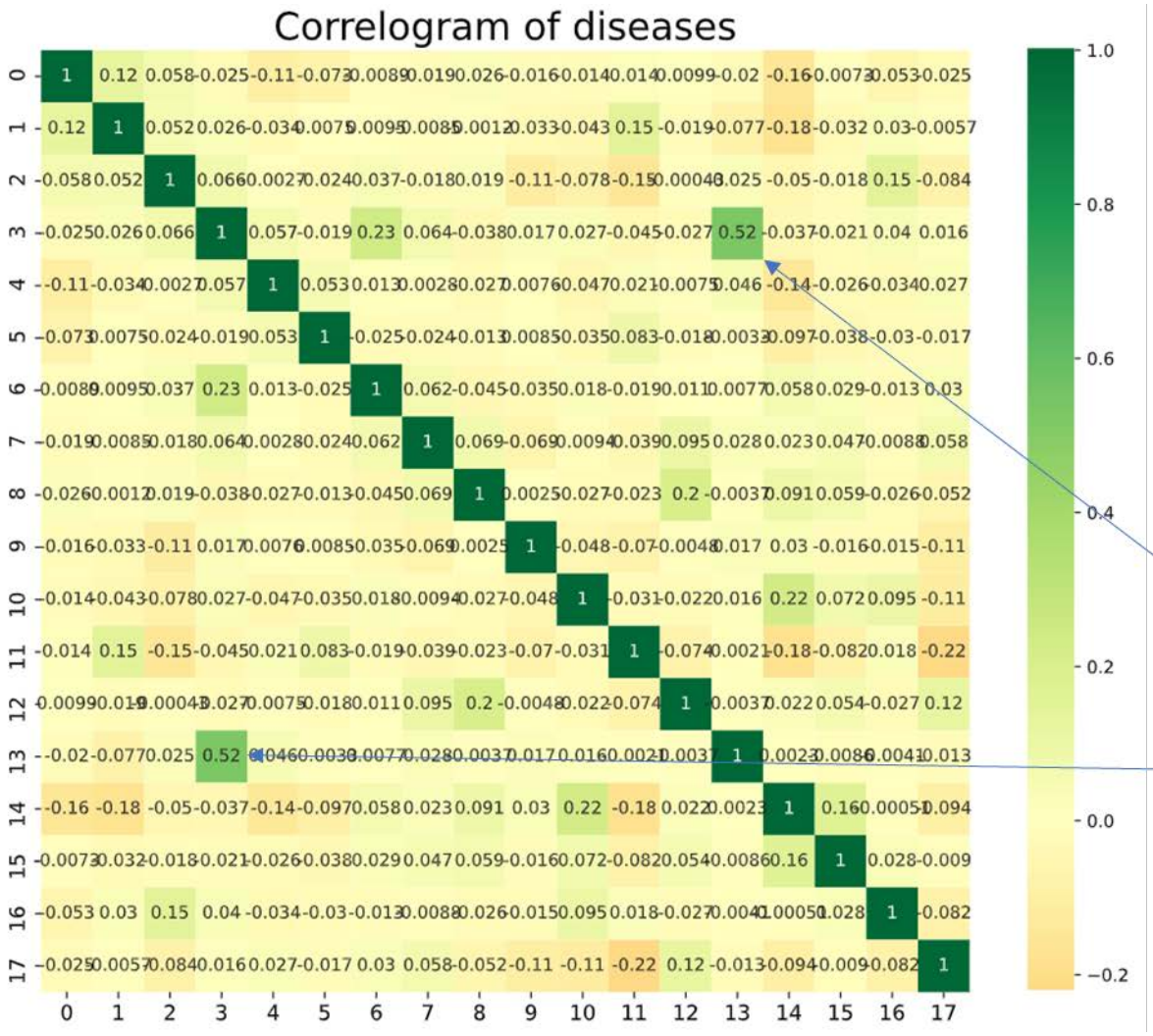

Figure 4: Correlogram of 18 most prevalent disease categories in the database. The numbers indicate disease name from fig.3 starting from "Sinus tachycardia", i.e. Sinus tachycardia"-> 0, "Left atrial enlargement"->1 ... Source code can be found in "Main_notebook.ipynb" in project code repository.[7]

Section 4: Results

To create a baseline, I first built a deep convolutional neural network which can classify the diseases in the database with very high accuracy. The architecture of the deep CNN is shown in Fig. 5.

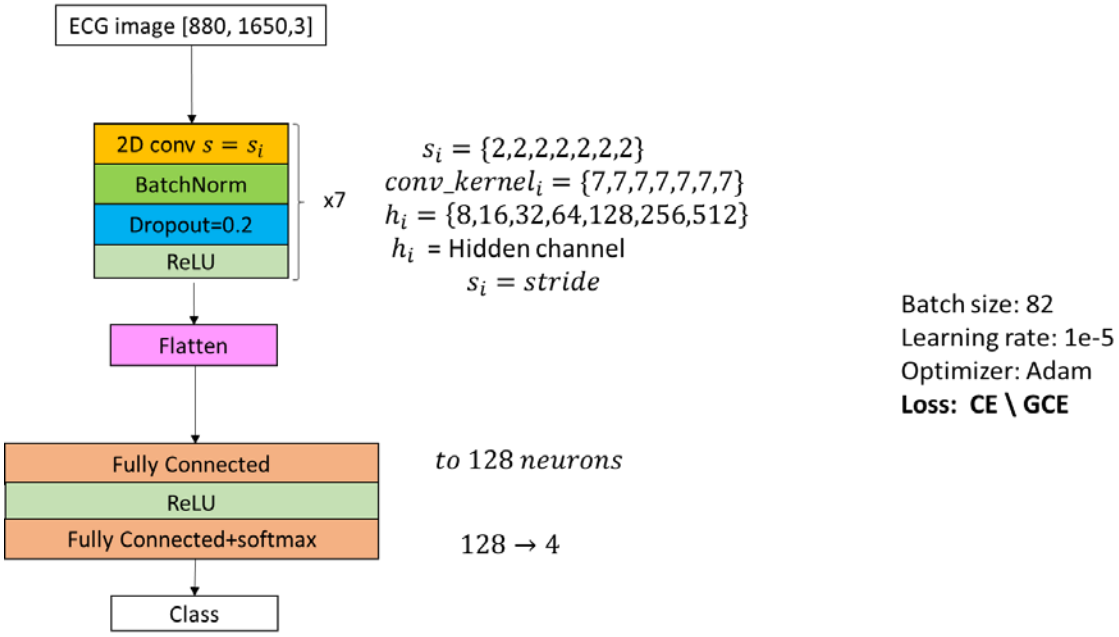

Fig. 5: Structure of the CNN capable of detecting cardiac disorders in the image with high accuracy.

The accuracy reached without treating the bias problem (baseline) is shown in Fig. 6.

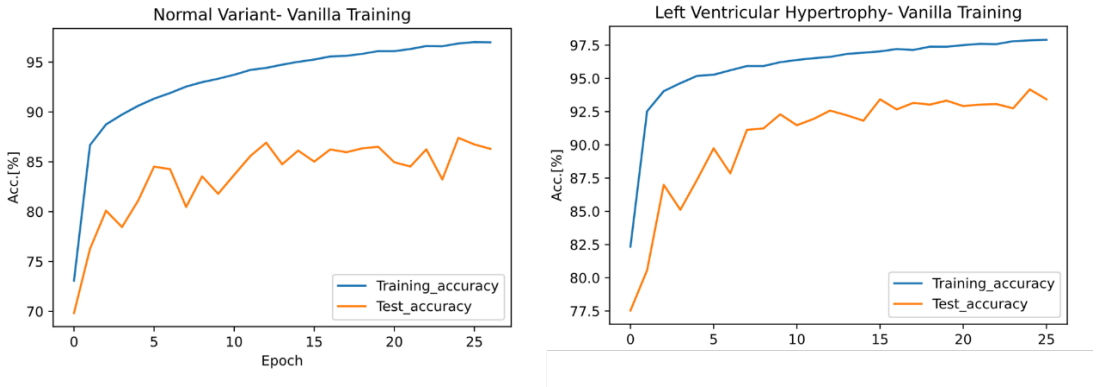

Fig. 6: Structure of the CNN capable of detecting cardiac disorders in the image with high accuracy. Source code can be found in "Main_script_net_training.py" in project code repository.[7]

After having this baseline, I used this to demonstrate the actual results of my extension and research:

(1) Behavior of the generalized cross entropy loss as expected and as described in Zhang et. al.[5] and mentioned in paper I chose[1]. I selected a disease of "Left ventricular hypertrophy" and trained the same net with various generalized cross entropy loss functions (different q values).

It was demonstrated empirically in the prior art[5] [1] that increase of q (amplification of generalization of cross entropy) can lead to significantly longer training time before convergence. Moreover, without the implicit weighting scheme to focus on challenging samples, the stochasticity involved in the training process can make learning difficult. As a result, classification accuracy might suffer. I proved it on physiological data, as can be seen at fig. 7.

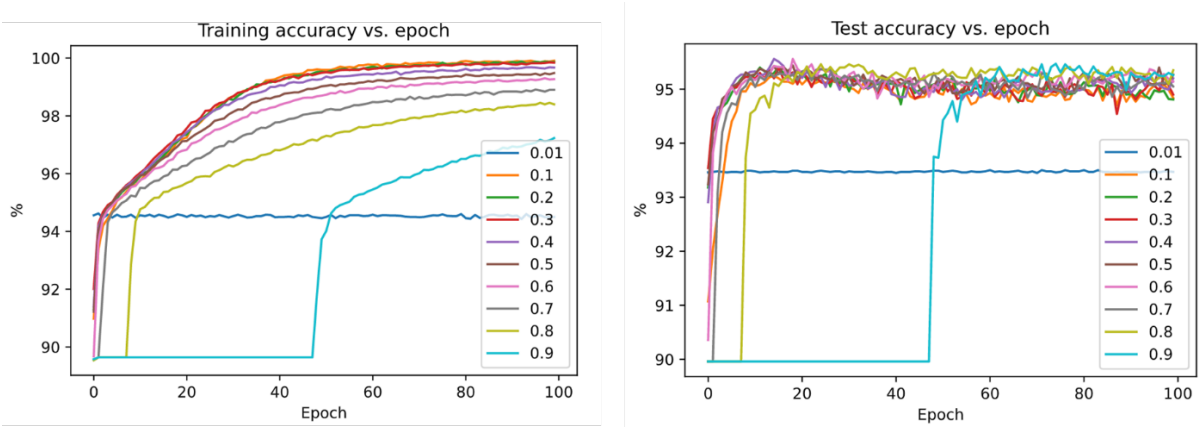

Fig. 7: Performance of the DNN from fig.5 with various generalized cross entropy finctions (various q values). Accuracy of the training set (left) and test set (right).

Indeed results shown if fig.7, confirm the prior art statement that as q value goes higher, the learning process more struggles to converge and reaches lower accuracy (at least on the training set). Another interesting insight that was not mentioned in the prior-art, but might be very beneficial is that generalized cross entropy actually helps avoid overfitting of deep neural networks with a large number of parameters (like dropout[8], for example). It can be seen from fig. 7 that the gap between the accuracy on the training set vs. accuracy on the test set is reduced, when q is increased. For very high q values, the perofrmance on both sets is almost equal!

(2) Assessment of effectiveness of method proposed by Nam et. al [1].

I implemented a method proposed by Nam et. al[1], applied it on the proposed database with different q values, and compared it to performance of vanilla net, i.e. net without debiasing. The code can be found on GitHub (see README file there)[9].

The results are shown at fig. 8.

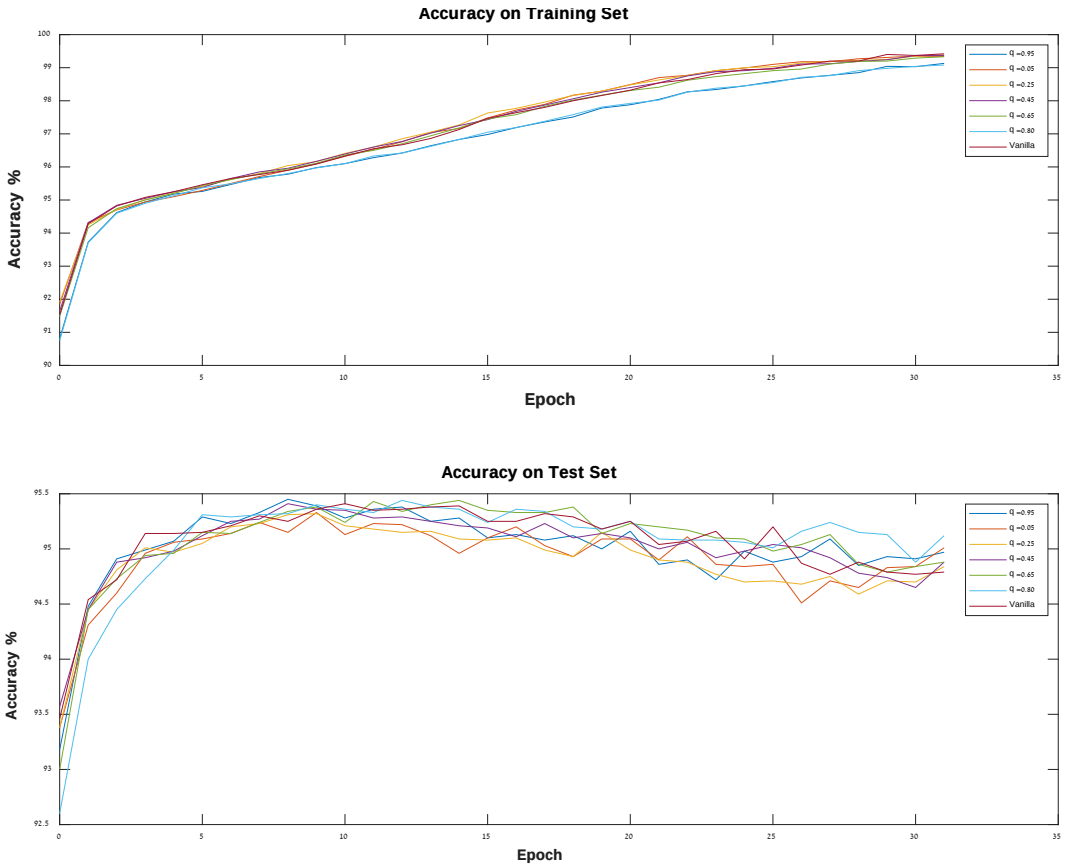

Fig. 8: Accuracy on a training set and on the test set as function of q, whereas

$$\mathrm{GCE}(p(x;\theta),y)=\frac{1-p_y(x;\theta)^q}{q}$$

As can be seen at fig. 8, In our case there is no significant contribution of this method to accuracy on the test set. Fig. 9 shows a zoom on a spot where test set accuracy reaches its maximum before overfitting. It can be seen there that although indeed nets trained with high q values reach slightly higher accuracy, the contribution is not very substantial.

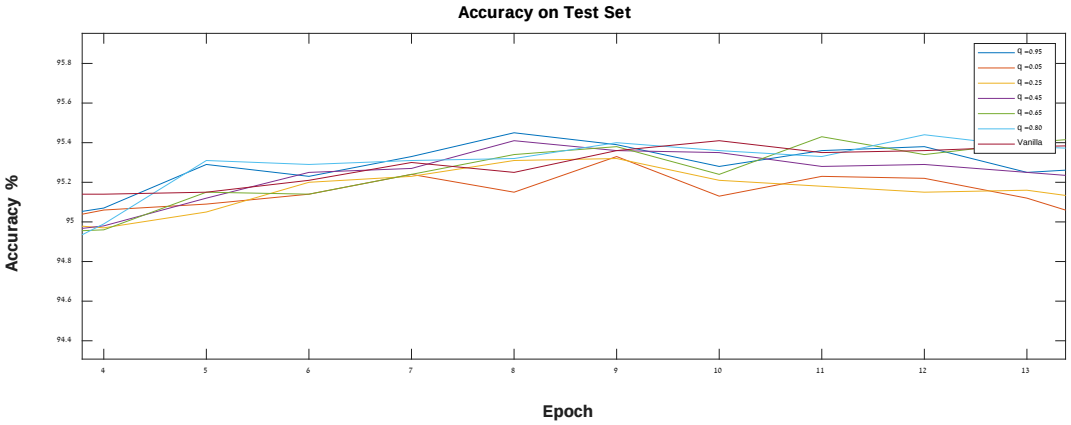

Fig. 9: Zoom on a maximal test set point. Although nets with higher q reach higher accuracy but the difference is not as substantial as mentioned in the prior-art.[1]

Section 5: Conclusion and future work

In conclusion, I proposed an additional assessment of behaviour of generalized cross entropy loss on physiological data. In addition, I found that it might be used not just as method for finding a biased samples, but also method for avoiding overfitting of deep neural networks. Additional achievement of this work is assement of proposed bias handling method on physiological database.

From my perspective, future research should focus on thorough examination of the approach to deeply understand which type of bias can be handled by the proposed method and which isn't, since I showed that the approach is not so effective in case of bias existing in my dataset. In addition, future reseach might focus on how to detect a bias which is treatable by the proposed method. Additional possible research directions are usage of not just cross entropy but other loss functions and its generalizations to point out bias samples and compensate for this bias during training.

References

1. Nam, J., Cha, H., Ahn, S., Lee, J. & Shin, J. Learning from Failure: Training Debiased Classifier from Biased Classifier. 1–19 (2020).
2. Kim, B., Kim, H., Kim, K., Kim, S. & Kim, J. Learning not to learn: Training deep neural networks with biased data. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2019-June**, 9004–9012 (2019).
3. Li, Y. & Vasconcelos, N. Repair: Removing representation bias by dataset resampling. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2019-June**, 9564–9573 (2019).
4. Wang, H., Xing, E. P., He, Z. & Lipton, Z. C. Learning robust representations by projecting superficial statistics out. *7th Int. Conf. Learn. Represent. ICLR 2019* 1–16 (2019).
5. Zhang, Z. & Sabuncu, M. R. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018-Decem**, 8778–8788 (2018).
6. NYU School of Medicine Emergency Care Electrocardiogram (ECG) Database. https://education.med.nyu.edu/ecg-database/app.
7. Gliner, V. Git repository of the project. https://github.com/vgliner/Learning_from_failure_final_project.
8. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
9. Gliner, V. Learning_from_failure_final_project. *GitHub* https://github.com/vgliner/Learning_from_failure_final_project (2021).