Biases in Machine Learning Models Using Medical Imaging
097248 Final assignment

Student name: Gal Peretz 201228525

October 2020

1 Introduction

Machine learning models are gaining popularity in recent years due to their capability of estimating almost any function given a large amount of training data. One of the use cases for machine learning models in the field of medical imaging is classifying MRI and CT scans to fully or partly automate the diagnosis process. The common paradigm is to train deep learning models in supervised fashion using labeled data that was already labeled by doctors. While the quantity of the data is important, the quality of the data is also affecting the performance of the models. If biased input is given in the training phase, the model could perceive the biased properties as dominant properties which could potentially lead to a classifier with poor performance. This can become a major problem when doctors depend on those models to decide the treatments of their patients. The training data often contain noise, because it is not created specifically for machine learning models but to help the doctors with the diagnosis task. In this paper, we will simulate the process of learning from baised data.

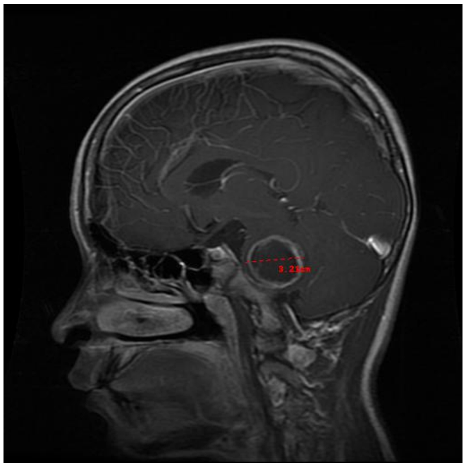


Figure 1: A small red line in the image that can easily be ignored by humans can cause problems for machine learning models. We need to force the model to ignore the red measurement line as a strong feature for classification.

2 Motivation

Machine learning models are based on data curated by humans, thus sensitive to any bias and noise that are hidden in the datasets. This can cause the model to learn from the wrong features. For example, a model that gets a good performance on skin cancer classification task only because doctors tend to measure the tumor size when they suspect that one exists. The model understands that, according to the training data, if the image contains a measurement line there is a good chance for cancer. Needless to say that this model is useless in real-world environments. The problem arises when the noise does not distribute equally between the examples, but instead has a higher probability for a specific type. In one hand the training data contains biased features that hard to remove, and this hurts the performance of the model. In the other hand we still want to learn useful features from this biased data and cannot afford to ignore it because the labeling process can be expensive especially in the healthcare space.

3 Problem Settings

Let $\mathcal{X}$ be a set of images and $\mathcal{Y}$ set of labels. Let $\mathcal{B}$ be a set of biased properties that $\mathcal{X}$ can possess with respect to $\mathcal{Y}$. We define $b : \mathcal{X} \to \mathcal{B}$, where $b(x)$ denotes the target bias of x. The problem that we want to solve is to learn a mapping function $F : \mathcal{X} \to \mathcal{Y}$ while unlearn the biased properties $\mathcal{B}$. To quantify how good our model minimize the effect of the biased properties $\mathcal{B}$, we will use the accuracy metric as our success metric.

4 Related work

The problem is similar to the one that Singh et al. (2020) dealt with in their paper. They used class activation maps (CAM) to automatically infer the location of different objects in the image to disentangle the object that is been recognized from the context of the image. The root of the problem is that standard models for image recognition often focus on the environment or the context of an object to recognize the object which lead to poor performance when dealing with different environment. Another approach that was suggested by Kim et al. (2019) is to learn to classify specific features in the image while learning to recognize the main object and then flip the gradient to unlearn the features that cause the bias. The architecture of this model consists of multiple convolution layers that extract features from the image and then continue to multiple classifiers. The first classifier tries to classify the main object and the others to classify the biased properties. Then the classifiers that classify the biased properties revere their gradient flow to affect the features extraction phase and to unlearn those biased features.Alvi et al. (2019) suggest mitigating the bias problem by using a primary classifier for learning the classification problem while simultaneously using additional classifiers to learn specific biased properties of the image and apply confusion loss that seeks to change the features representation of the main task such that it becomes invariant to biased properties. The confusion loss is just a cross entropy loss between the classifier output and uniform distribution. Another interesting approach explored by Chen et al. (2016) is using a generative adversarial network (GAN) introduced by Goodfellow et al. (2014) to learn disentangled representation of the features.

5 Dataset

To simulate real world situations we will use dataset of MRI scans with 3 types of brain tumors created by Cheng (2017). we will inject bias in the images according to the type of the tumor. glioma brain tumors are tumors that form on the supportive tissue of the brain which makes them more dangerous than other brain tumors thus we assume that doctors would want to measure the tumor size often to monitor the growing rate and to understand the urgency of surgery. We add to these images with probability of ρ a red measurement line to simulate the process of helping the doctor with the diagnosis process. Pituitary tumors are more rare tumors that are located in the pituitary. the pituitary is a small gland that sits inside the skull and produces hormones that regulate the levels of other hormones giving it an important role in the hormonal system. For images that contain pituitary tumors, we added with probability of γ a blue polygon that measures the area of the tumor to simulate the process of approximate the effect of the tumor on the hormonal system. The last type of brain tumor is meningioma. Most of the time these tumors are benign and some meningiomas may not need immediate treatment and may remain undetected for years. For this type, we didn't change the original images.

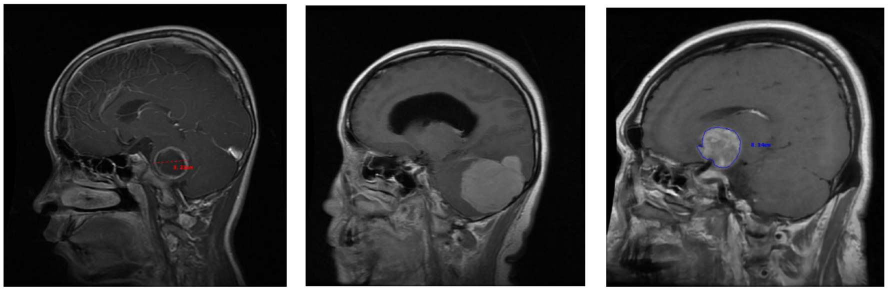


Figure 2: Data sample that contains the different biases for each label that were injected to the original dataset. the label bias for the giloma cases is a red line that measure the tumor size and for the pituitary is a blue polygon around the tumor that measure the area.

6 Model

In this paper, we will test the architecture suggested by Kim et al. (2019) to test if it also works on dataset that simulates a real world situations. The model that was suggested by this paper consist of three sub-models. The first one is the features extractor model. the output of this model flows to the image classifier model to predict the label and to the bias predictor model to predict the bias in a specific image.

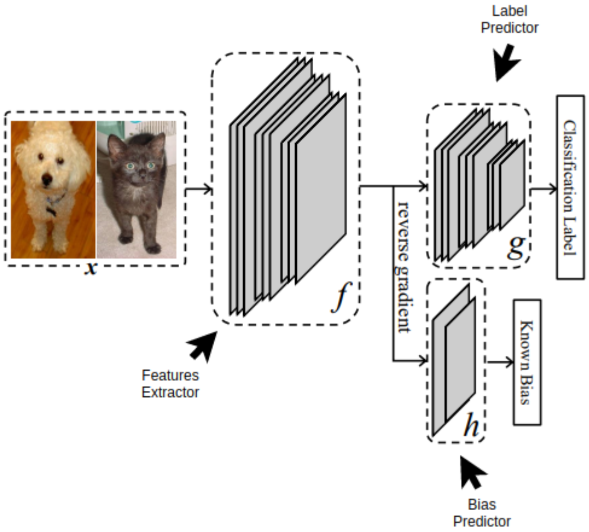


Figure 3: Original model architecture

This model architecture was implemented using ResNet-18 to compute $g \circ f$ while also flowing f's output to h. The h function consists of multiple convolution layers with relu activation to predict the bias.
It is important to note that the authors assume that, the biased properties are known in advance. The bias classifier can help the model to ignore the biased properties and to use only the relevant features to classify the tumor type.

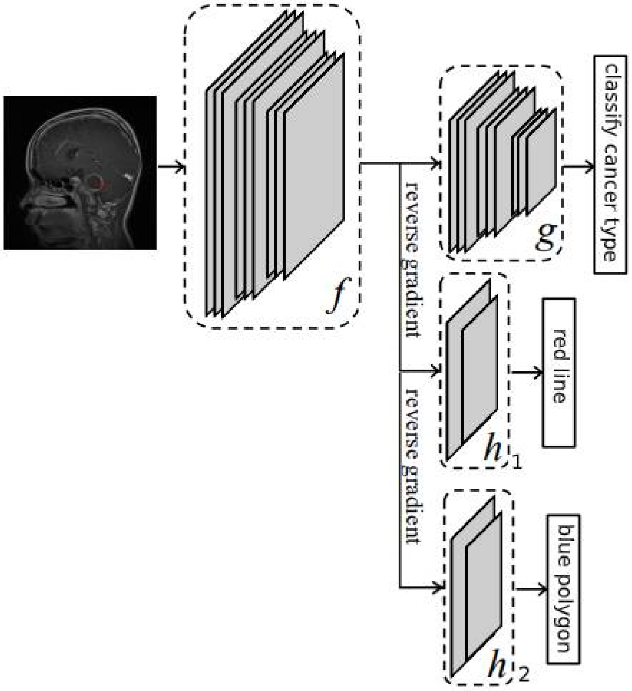


Figure 4: The suggested model

Our model will be composed of multiple convolution and relu activation layers for features extraction and with three additional classifiers each will be composed of multiple convolution and relu activation layers. The first one will try to classify the type of the tumor. The second will try to classify if the red measurement line exists in the image and the third will try to classify if the blue polygon exists in the image. Then we will reverse the gradient flow of the second and third models to unlearn the biased properties.

7 Training

To train the model we will use cross entropy loss to train the primary classifier and a binary cross entropy to jointly train the label predictor and the bias predictors. Instead of backpropgate the gradient from the bias predictors we will use a reverse gradient layer to flip the gradient coming from the h functions.

8 Experiments and Results

Humans can easily overlook small details like the red measurement line and the blue polygon and count them as insignificant. Either, because we can miss those small details in the image or because we, as humans, understand that those details should not affect the tumor type and ignore them.
To check the effects of those allegedly insignificant adjustments to the scans on machine learning models we use a ResNet18 classifier as the feature extractor and try to classify the tumor type with and without the adjustments to the training data.

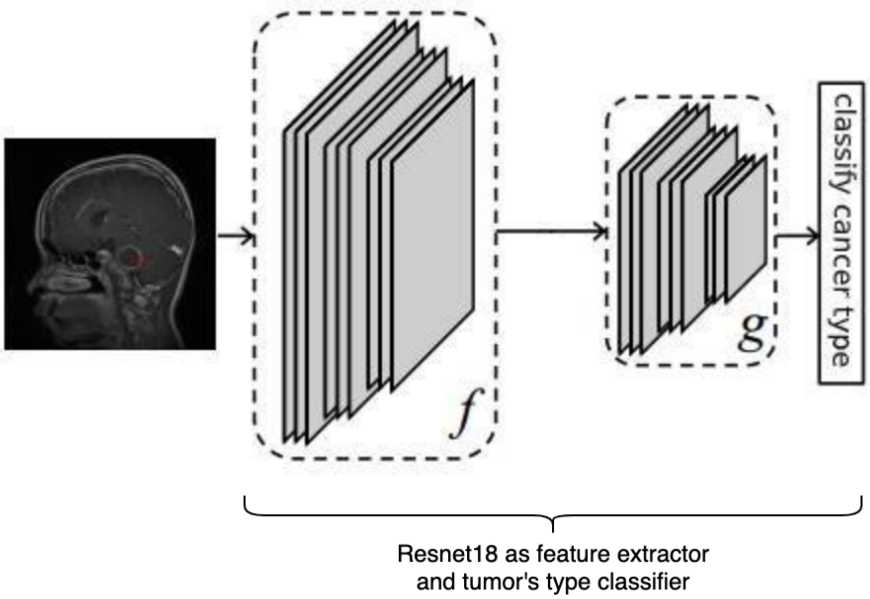
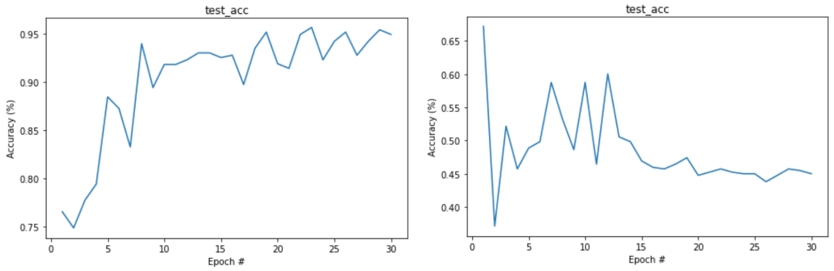


Figure 5: Standard classifier

To check if standard model can handle the biased properties. we used ResNet18 and split it to feature extractor model and classifier model. then we check the performance of the model with and without the biased properties.



(a) Without the biased properties

(b) With the biased properties

The max accuracy of the classifier given the original data as the training data is 0.96 while the accuracy of the classifier using the adjusted version when $\rho = \gamma = 1$ is 0.62. This means that unlike humans machine learning models cannot understand that these properties are misleading which causes a large drop in performance. The next experiment checks if the model can ignore the biased properties if those exist only in part of the images. For that, we add only the red measurement line to the glioma tumors with probability ranging from 0.5 to 1.

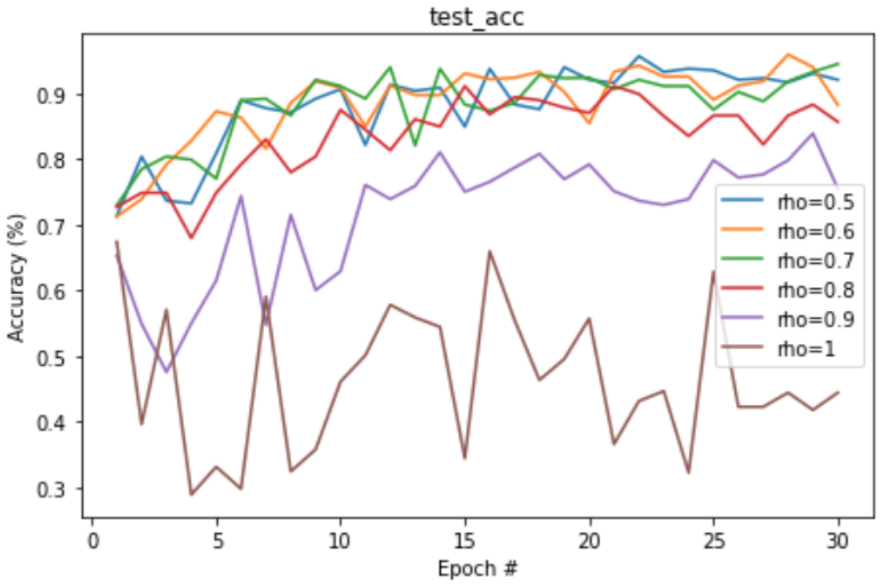
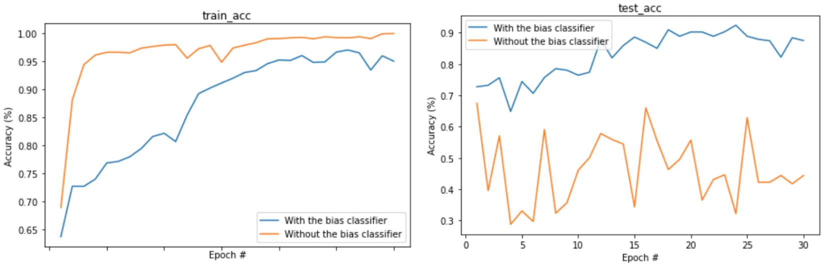


Figure 7: Performance of the classifier with different ρ values ranging from 0.5 to 1

Model/Dataset	$\rho = 0.5$	$\rho = 0.6$	$\rho = 0.7$	$\rho = 0.8$	$\rho = 0.9$	$\rho = 1$
ResNet18	0.957	0.959	0.944	0.911	0.838	0.6739

Table 1: The max accuracy of the classifier for different rho values.

We can see that the classifier can learn to ignore the red line if it appears in less than 80 percent of the images containing the glioma tumors. However, when the prevalence of it increases the performance of the model decreases dramatically. In the next experiment we will check if the architecture suggested by Kim et al. (2019) can mitigate the effect of those biased properties. First, we will check if it can mitigate the effect of just one property, and then we will see if it can scale up to multiple properties. In the next experiment, we will use another classifier that share the same feature extractor to spot the red measurement line in the images and then reverse the gradient to force the feature extractor to ignore this property.



(a) accuracy compression for train

(b) accuracy compression for test

Form this experiment we can learn that the extra classifier can help the model to gain a better accuracy. The max accuracy of the model with the extra classifier is 0.9231 while the max accuracy of the version without it is 0.6739. The graph of the train process emphasizes that the model without the extra classifier learns quickly to depend on the red line and gets to a local minimum after 5 epochs while the suggested model learning curve is more steady which mean that it learns more useful features and ignore the biased properties. The next thing we can check is the scalability of the architecture. For that, we will add another classifier to deal with the blue polygon bias.

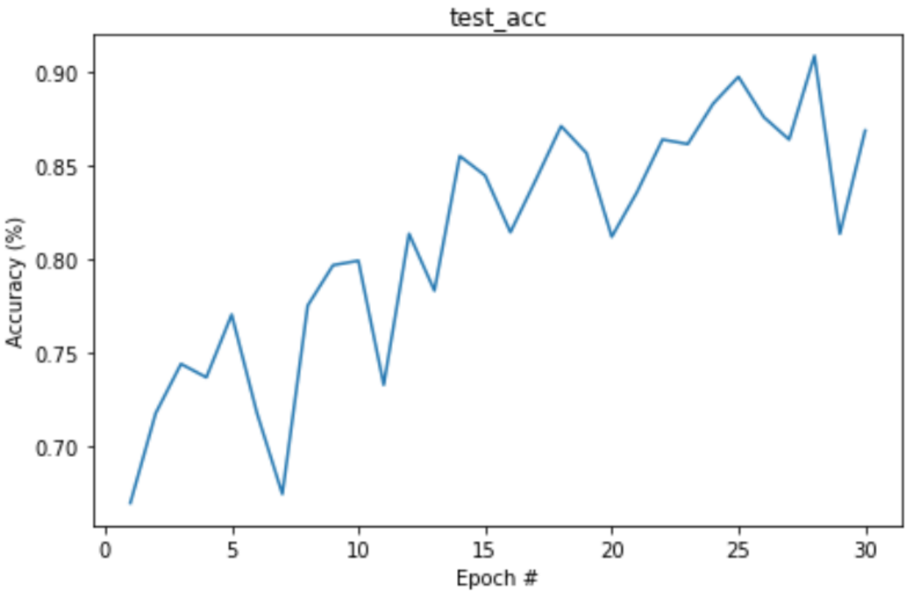


Figure 9: Accuracy of the model with the architecture model using the two extra classifiers

We can see that the model with the extra classifiers can perform well even with two biased properties getting an accuracy of 0.91 compared to the 0.67 that the model without the extra classifiers got with the same settings. This indicates that the suggested architecture by Kim et al. (2019) is also scalable. However, it does suffer from performance drop compare to the non-bias version and also compare to the model that deals with single bias.

9 Limitations of the study

In this paper, we tried to understand if the architecture suggested by Kim et al. (2019) works well on real-world dataset of MRI scans that may contain noise and biased properties. Unfortunately, we don't have any access to such dataset, so we were forced to modify an existing dataset to simulate the situation. In the discussion section, we will talk about suggestions for dealing with the problem of unknown biased properties using unlabeled data. Since we don't have access to large amount of unlabeled MRI scans we cannot check the effect of those suggestions.

10 Discussion

In this paper, we explored the problem of dealing with data that contain noise and label bias. We can understand that if we know the bias in advance then the architecture suggested by Kim et al. (2019) can help mitigate it's effects. However, the assumption that biased features are known in advance does not always hold in real-world situations. Often, biased dataset is a result of the fact that labeling the training data is an expensive and time-consuming operation which leads to label only subset of the data that does not represent the overall population good enough. In contrast to the labeling task, getting unlabeled data is usually an easier and less expensive task, therefore we can get enough examples that represent the distribution of the population better than the labeled dataset. Intuitively, we can use autoencoder Baldi (2012) architecture to estimate the right features that can reconstruct the different images in the unlabeled dataset and combine it with the current model to regularize the biased features in the training phase. In conclusion, I think that the architecture suggested by Kim et al. (2019) can help us deal with known bias properties but further research is needed to deal with unknown biases.

References

Alvi, M., Zisserman, A., and Nellåker, C. (2019). Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. pages 556–572.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. pages 37–49.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. page 2180–2188.

Cheng, J. (2017). brain tumor dataset.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. pages 2672–2680.

Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. (2019). Learning not to learn: Training deep neural networks with biased data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9012–9020. Computer Vision Foundation / IEEE.

Singh, K. K., Mahajan, D., Grauman, K., Lee, Y. J., Feiszli, M., and Ghadiyaram, D. (2020). Don't judge an object by its context: Learning to overcome contextual bias. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11067–11075. IEEE.