Deep CNN for Medical Image Classification

By: Mohamed Stouka, Pei Ge, Obaid Elshafiey

Michigan State University

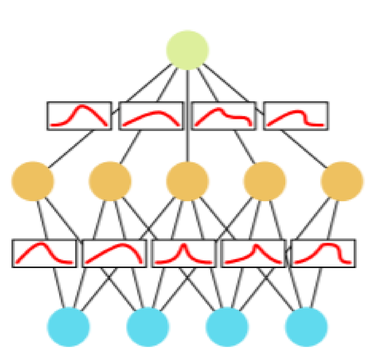
Introduction

Medical image classification plays an important role in clinical treatment and teaching. However, traditional methods can be time consuming, and in some fields, have reached their ceiling on performance. Effectively classifying medical images can be utilized in aiding clinical care and treatment and can reduce the diagnosis process's length significantly. In recent years, Convolutional Neural Networks (CNNs) have been utilized for diagnosing diabetic retinopathy (DR) through analyzing fundus images and have proven their superiority in detection and classification tasks. For diabetes, DR is a major complication that may eventually result in vision loss as well as blindness. In this project, we propose a novel deep CNN architecture that can classify subjects into 4 levels of disease severity; Healthy (i.e. no DR), moderate DR that includes patients with mild or moderate NPDR, and severe DR, which represents patients in the late stages with either severe NPDR or PDR. The proposed architecture was trained and tested on a sample fundus images generated from a publicly available dataset from Kaggle , the dataset size was of approximately 35,000 images.

Objectives

Deep learning models are prone to overfitting, which negatively affects their generalization capabilities, they also tend to be overconfident about their predictions (when they do provide a confidence interval). All of this is problematic for applications such medical diagnostics [3]. Research in recent years put forward many approaches that can mitigate these drawbacks especially via the use of bayesian neural networks to estimate the uncertainty in the model prediction. In this project, we follow this approach to aid us in dealing with the difficult nature of the images as well as the unbalanced dataset to not only predict the class of the disease severity, but also how much we are uncertain of our prediction.

A common definition is that a Bayesian neural network is a stochastic artificial neural network trained using Bayesian inference [4]



As shown in the illustrative figure above, the values for the weights and biases, unlike in point estimates networks, are probability distributions that we can sample from to generate a traditional neural network.

The main goal of using a stochastic neural network architecture is to get a better idea of the uncertainty associated with the underlying processes. This is accomplished by comparing the predictions of multiple sampled model parametrization θ. If the different models agree the uncertainty is low. If they disagree, then it is high. This process can be summarized as follow [4]:

$$\theta \sim p(\theta),$$
$$y = N\, N_\theta(x) + \epsilon,$$

Where ϵ represents random noise to account for the fact that the function N N is just an approximation.

When a BNN for this project, we had to main steps to accomplish, the first step was the choice of a deep neural network architecture. Then, we had to choose a stochastic model, i.e., a prior distribution over the possible model parametrization (p(θ)) to train the dataset on.

Methodology

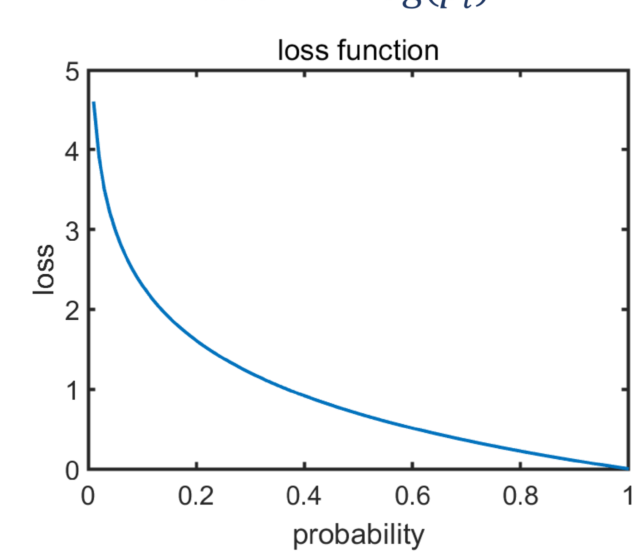
1. Model Structure: We use Conv2DFlipout layer and DenseFlipout [1] layer from TensorFlow-Probability to get the probability distribution. We also add a Dropout layer to avoid overfitting.

Layer (type)	Output Shape	Param #
conv_tfp_1a (Conv2DFlipout)	(None, 256, 256, 16)	2416
max_pooling2d (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d (Conv2D)	(None, 126, 126, 128)	18560
max_pooling2d_1 (MaxPooling2D)	(None, 63, 63, 128)	0
conv_tfp_1b (Conv2DFlipout)	(None, 63, 63, 32)	73760
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 30, 30, 128)	36992
max_pooling2d_3 (MaxPooling2D)	(None, 15, 15, 128)	0
flatten (Flatten)	(None, 28800)	0
dense (Dense)	(None, 1024)	29492224
dense_1 (Dense)	(None, 512)	524800
dropout (Dropout)	(None, 512)	0
dense_flipout (DenseFlipout)	(None, 5)	5125

2. Loss Function: Negative Log Likelihood

$Y_{pred} \in R^n$ is a vector $(p_1, p_2, \cdots, p_n)$ where p_i means the probabilities of the i^{th} class. For an image in i^{th} class,

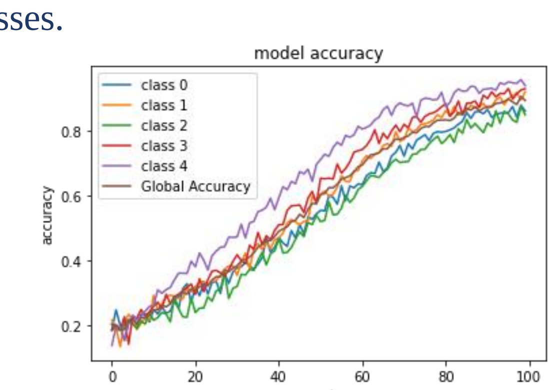
$$loss = -\log(p_i)$$



The benefit of this loss function is that it's nonlinear, so the low probability gives its large weight when minimizing the loss.

3. Metrics

The accuracy is defined as the number of correct predictions divided by the total number of predictions. We apply this accuracy to all the 5 class respectively and as well as the data set. This can help us check if the model works on all 5 classes.



4. Balance the Dataset

Our dataset is not balanced, as the class 0 consists of more than 70% of samples. So, the model will ignore other classes, and just classify all the image as class 0. This gives an accuracy of 70% which is not acceptable.

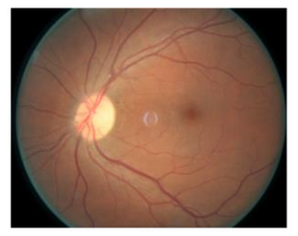
Class	0	1	2	3	4
Number of samples	25802	2438	5288	872	708

To balance the dataset, we delete most of the data set and keep of the class have 700 images. Then we use data augmentation to enlarge it.

Results and Discussion

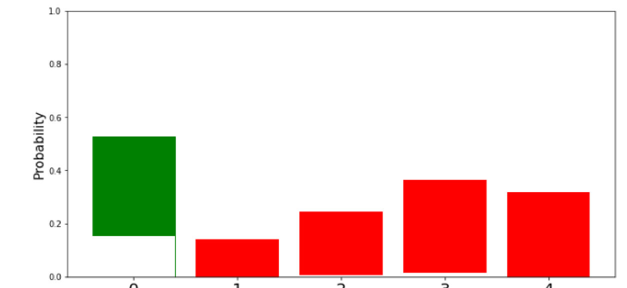
1. In this project we created a Bayesian neural network model to quantify the uncertainty in Diabetic Retinopathy dataset, the concepts developed are also applicable in other medical datasets

- The Bayesian method used provided us with a natural way to calculate the two types of uncertainty: Aleatoric uncertainty and epistemic uncertainty
- For calculating the uncertainty, we use methods developed in (Kwon et al., 2020) in which the uncertainty is driven form the variance of the prediction probability of the neural network
- Using an approximative method, we can achieve high accuracy with low computational cost.

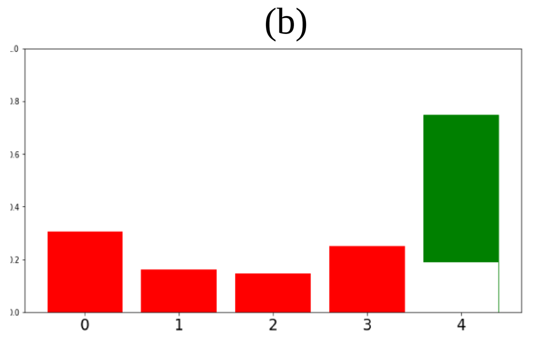


(a)

(b)



(c)



(d)

2. Explaining the results:

From the figure above

(a) shows a fundus eye image with no signs of diabetic retinopathy

(b) shows an image with extreme signs of DR

(c) Displays the prediction accuracy for the 4 classes for the fundus image with no DR signs, the model correctly assigns the image to class 0.

(d) Displays the prediction accuracy for the 5 classes for a fundus image with high proliferative signs of DR.

3. Calculating the uncertainty

According to the method presented in [2] the uncertainty correlates with the prediction variance as shown the equation below:

$$\mathrm{Var}_{P(\omega|D)}(y) = Al + Ep$$

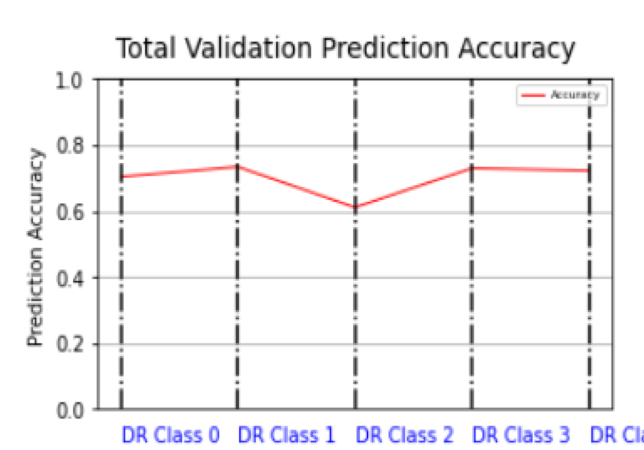
Where $\mathrm{Var}_{P(\omega|D)}(y)$ is variance of the prediction accuracy Al is the Aleatoric, uncertainty and Ep is the Epistemic uncertainty.

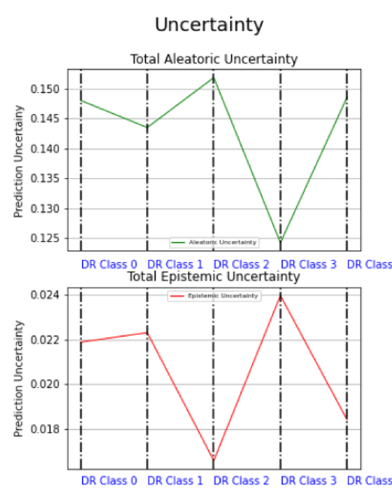
Al and Ep can be calculated as follows:

$$Al = \frac{1}{N}\sum_{n=1}^{N} diag\{p(y|x,\omega_n)\} - p(y|B_z,\omega_n)^{\otimes 2}$$

$$Ep = \frac{1}{N}\sum_{n=1}^{N} \{p(y|B_z,\omega_n)\} - p(y|B_z)\}^{\otimes 2}$$

4. Performance of the algorithm is displayed by averaging the accuracy and the uncertainty for each class





Conclusion

- We proposed a new method to quantify the uncertainty in classification using the relation between the variance and mean (avoiding the estimation of extra parameters).
- Compared to literature, our model proves to be less resource demanding for similar accuracy. The Calculated uncertainty is a useful insight in understanding how the model trains and can assist us in future improvements.

References

[1] Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018). *Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches*. https://arxiv.org/abs/1803.04386

[2] Kwon, Y., Won, J. H., Kim, B. J., & Paik, M. C. (2020). Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics and Data Analysis*, *142*, 106816. https://doi.org/10.1016/j.csda.2019.106816

[3] J. Ker, L. Wang, J. Rao, and T. Lim. Deep learning applications in medical image analysis. IEEE Access, 6:9375–9389, 2018

[4] L. Jospin, W. Buntine, F. Bpussaid, H. Laga, M.Bennamoun (2020). *Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users*, arXiv:2007.06823v1

Acknowledgements

This is the final project of ECE884 of Michigan State University. Many thanks to Professor Mi Zhang for his guidance throughout the entire semester.

Contacts

Mohamed Stouka: stoukamo@msu.edu

Pei Ge: gepei@msu.edu

Obaid Elshafiey: elshafi1@msu.edu

GitHub repo: https://github.com/qdgp/Project-of-ECE884