Leveraging uncertainty information from deep neural networks for disease detection (Leibig et al. 2016)

Valerie Ding, Thaminda Edisooriya, Samir Sen, Meera Srinivasan

Background

- CNNs
- Dropout
- Bayesian Estimation via Dropout

Introduction

- DNNs - State of the Art in many disease detection problems
- Not much exploration into the uncertainty of results
- Want to be able to refer hard cases to human experts
- Solution: Bayesian Estimates of uncertainty

Introduction - Data

- Diabetic Retinopathy - blindness caused by blood sugar levels in blood vessels in the eye being damaged
- Predict disease based on images of the back of the eye
- Two datasets, with varying difficulty - Kaggle challenge dataset, and Messidor database

Introduction - Measuring Uncertainty

Approximate Bayesian Inference of Uncertainty using Monte Carlo Dropout

Compares against:

- straightforward dropout and softmax score
- Gaussian process run on last few layers of network

Calculation of Uncertainty

Dropout samples a random set of points from each layer in the convolution layer to the next. => Store this result as an example.

By taking 100 simulations of the dropout classification task and computing the average we get the U_pred and SD_pred.

Without dropout turned on, the Monte Carlo simulation would result in the same score as we are just optimizing the same objective function

Dropout allows to sample the prediction posterior distribution and calculate the associated statistics

Introduction - Brief Results

- State of the Art performance on the Messidor dataset
- Can use uncertainty to refer hard cases to authority
- Better performance on more certain cases
- Conventional network output probabilities DO NOT capture uncertainty well

Methods

- CNN → Disease Prediction
- Monte Carlo Dropout → Uncertainty Measure
- Gaussian Processes

Methods - CNN

- JFNet - 13 convolutions, 5 max pools, 3 fully-connected at the end
  - Dropout (P=0.5) between all fully connected layers
  - ReLU or leaky ReLU after all weight layers
- Trained using SGD with Nesterov Momentum
  - 30 epochs, piecewise constant learning rate
  - L2 regularization on all weights, L1 on final layer
- Data augmentation
  - Affine transforms applied to input images, 10 deg rotation, 25 pixel shift
  - Randomly flip images across either axis

Methods - Monte Carlo Dropout

- T = 100 samples per test point
- Compute mean, standard deviation of softmax probabilities for each sample
- Dropout set to (P=0.2), layers scaled accordingly

Results

- Disease Detection Rates with Referral on “hard” cases
- Monotonically Increasing accuracy and confidence
- Significant increase in confidence and accuracy after rejecting a small percentage of “hard” examples
- Less confident on “healthy” examples - it’s more obvious when an example is clearly diseased, but less so when an example is healthy but looks different

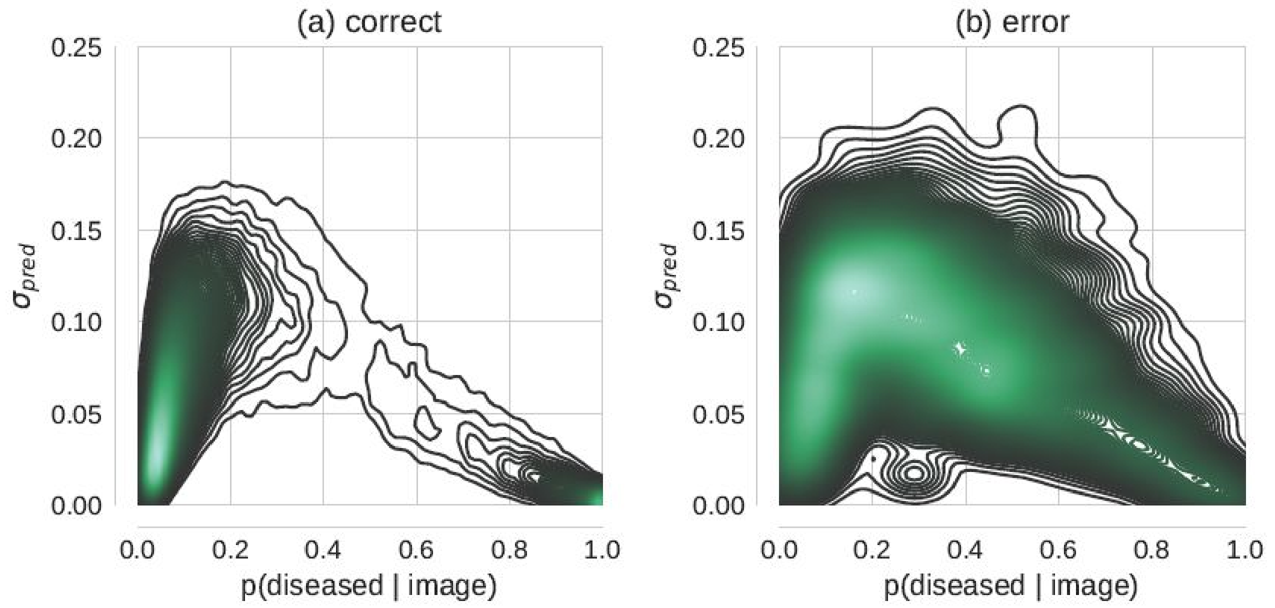

Figure 2. Relation between Bayesian model uncertainty σ_{pred} and maximum-likelihood, i.e. conventional softmax probabilities $p(diseased|image)$. Each subplot shows the 2-dimensional density over Kaggle DR test set predictions conditioned on: correctly **(a)** vs. erroneously **(b)** classified images respectively.

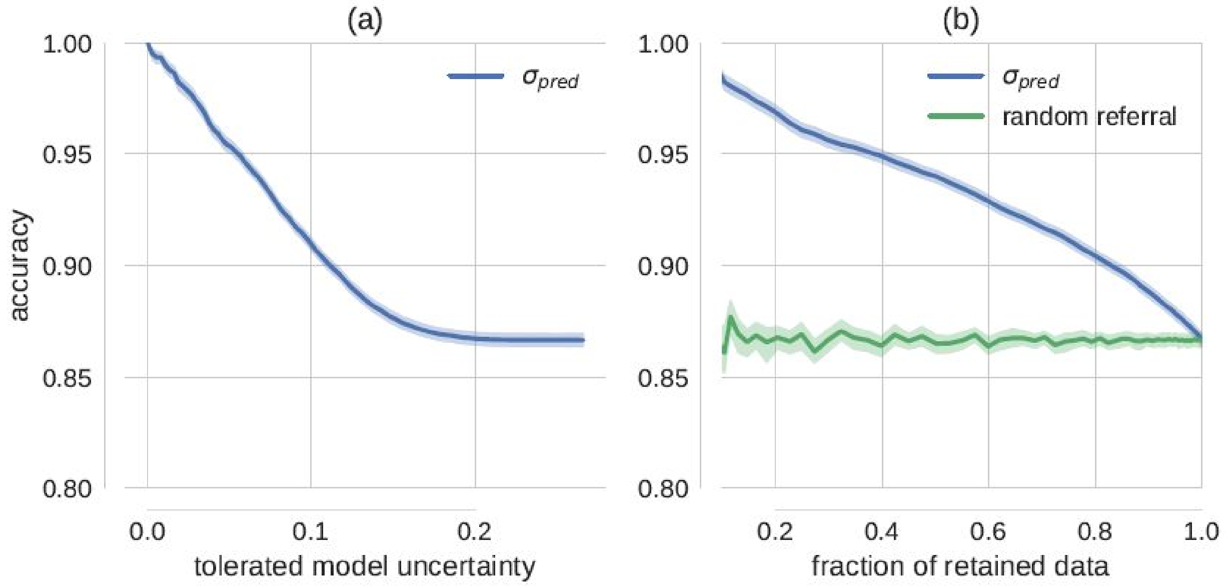

Figure 3. Improvement in accuracy via uncertainty-informed decision referral. **(a)** The prediction accuracy as a function of the tolerated amount of model uncertainty. **(b)** Accuracy is plotted as a fraction of retained data. The green curve shows the effect of rejecting the same number of samples randomly, that is without taking into account information about uncertainty.

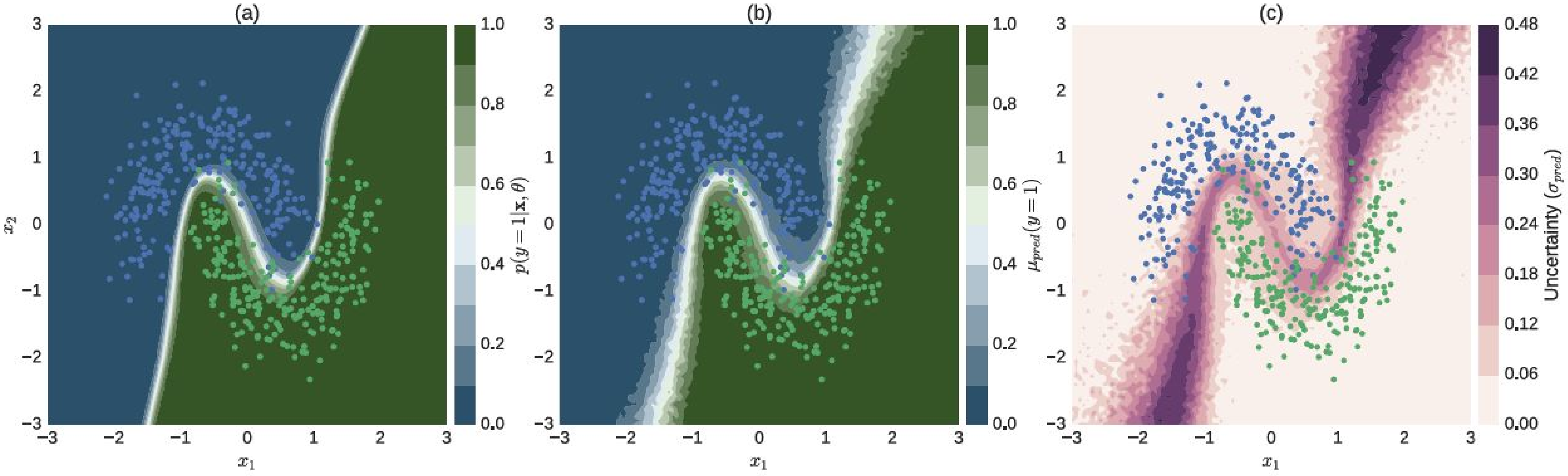

Figure 5. Illustration of uncertainty for a 2D binary classification problem. **(a)** Conventional softmax output obtained by turning off dropout at test time (eq. 1). **(b)** Predictive mean of approximate predictive posterior (eq. 7). **(c)** Uncertainty, measured by predictive standard deviation of approximate predictive posterior (eq. 8). The softmax output (a) is overly confident (only a narrow region in input space assumes values other than 0 or 1) when compared to the Bayesian approach (b, c). Uncertainty (c) tends to be high for regions in input space through which alternative separating hyperplanes could pass. Colour-coded dots in all subplots correspond to test data the network has not seen during training.

Discussion

- Runs quickly and is thus still useful - 200ms per test example
- Performance of the classifier is state of the art on one dataset, even though model was not explicitly tuned as part of this experiment
- Worse performance on Kaggle dataset - more empirical uncertainty in the dataset as evidenced by expert opinion
- Explicit uncertainty estimation outperforms uncertainty based on softmax scores

Limitations

- Uncertainty approach in discriminative model not well-suited for detecting adversarial attacks or handling “far out” data
- “Multiple hyperplanes” wording
  - Debatable description of new B-CNN uncertainty models
- Kaggle dataset concerns, indicative of model performance dependency on data and questions re: uncertainty model applicability

Thank You