Who Said What: Modeling
Individual Labelers Improves Classification

Melody Y. Guan[*]

Stanford University
450 Serra Mall
Stanford, California 94305
mguan@stanford.edu

Varun Gulshan, Andrew M. Dai,
Geoffrey E. Hinton

Google Brain, 1600 Amphitheatre Pwky
Mountain View, California 94043
{varungulshan, adai, geoffhinton}@google.com

Abstract

Data are often labeled by many different experts with each expert only labeling a small fraction of the data and each data point being labeled by several experts. This reduces the workload on individual experts and also gives a better estimate of the unobserved ground truth. When experts disagree, the standard approaches are to treat the majority opinion as the correct label and to model the correct label as a distribution. These approaches, however, do not make any use of potentially valuable information about which expert produced which label. To make use of this extra information, we propose modeling the experts individually and then learning averaging weights for combining them, possibly in sample-specific ways. This allows us to give more weight to more reliable experts and take advantage of the unique strengths of individual experts at classifying certain types of data. Here we show that our approach leads to improvements in computer-aided diagnosis of diabetic retinopathy. We also show that our method performs better than competing algorithms by Welinder and Perona (2010); Mnih and Hinton (2012). Our work offers an innovative approach for dealing with the myriad real-world settings that use expert opinions to define labels for training.

Introduction

Over the last few years, deep convolutional neural networks have led to rapid improvements in the ability of computers to classify objects in images and they are now comparable with human performance in several domains. As computers get faster and researchers develop even better techniques, neural networks will continue to improve, especially for tasks where it is possible to get a very large number of accurately labeled training examples. In the near future, we can expect neural networks to start serving as alternatives to human experts. We would, in fact, like the neural networks to perform much better than the human experts used to provide the training labels because these training labels are often unreliable as indicated by the poor agreement between different experts (55.4% for the datasets we consider) or even between an expert and the same expert looking at the same im-
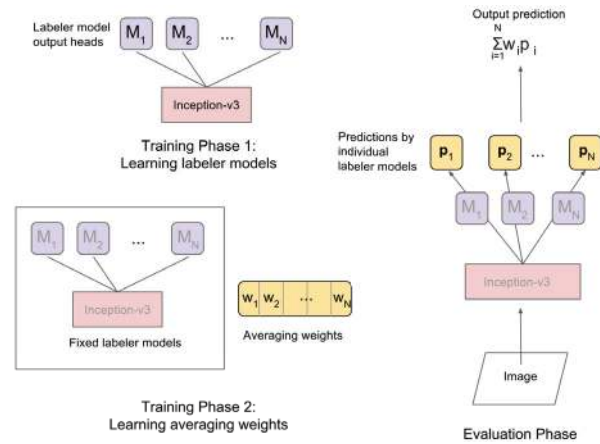

Figure 1: Overview of proposed procedure to learn classification from multiple noisy annotators.

age some time later (70.7%).[1] Intuitively, we would expect the quality of the training labels to provide an upper bound on the performance of the trained net. In the following section we show that this intuition is incorrect.

Our paper's main contribution is to show that there are significantly better ways to use the opinions of multiple experts than simply treating the consensus of the experts as the correct label or using the experts to define a probability distribution over labels.

Figure 1 summarizes our optimal procedure.

Beating the Teacher

To demonstrate that a trained neural net can perform far better than its teacher we use the well-known MNIST handwritten digit benchmark for which the true labels are known and we create unreliable training labels by corrupting the true labels. This corruption is performed just once per experiment, before training starts, so the noise introduced by the corruption cannot be averaged away by training on the same example several times. MNIST has 60k training images and 10k test images and the task is to classify each

[1]Inter-grader variability is a well-known issue in many settings in which human interpretation is used as a proxy for ground truth, such as radiology and pathology (Elmore et al. 1994; 2015).
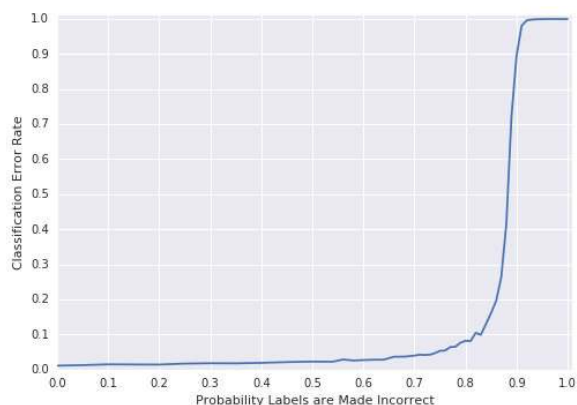
Figure 2: Performance of a deep neural net when trained with noisy labels.

(a) Healthy (b) Mild NPDR

(c) Moderate NPDR (d) Severe NPDR

(e) Proliferative DR

Figure 3: Sample fundus images from each DR class.

28×28-pixel image into one of the ten classes. For the purposes of this demonstration, we use a very simple neural net: two convolutional layers with 5×5 filters, rectified linear unit (ReLU) activation functions, and 16 and 25 output channels respectively, each followed by a max pooling layer with 2x2 filters; a fully connected hidden layer of 32 ReLUs; and a 10-way softmax layer. We train the net on 50k examples using stochastic gradient descent on mini-batches of size 200 with the Adam optimizer (Kingma and Ba 2015) and we use the remaining 10k training cases as a validation set for tuning the learning rate and the magnitude of the initial random weights. The best-performing net has a test error rate of 1.01% when the training labels were all correct. If the labels are corrupted by changing each label to one of the other nine classes with a probability of 0.5, the test error rate only rises to 2.29%. Even if each training label is changed to an incorrect label with probability 0.8 so that the teacher is wrong 80% of the time, the trained net only gets 8.23% test error. If the teacher is even less reliable there comes a point at which the neural net fails to "get the point" and its error rate rises catastrophically, but this does not happen until the teacher is extremely unreliable as shown in Figure 2.

This demonstrates that the performance of a neural net is not limited by the accuracy of its teacher, provided the teacher's errors are random. One obvious question is how many noisily labeled training examples are worth a correctly labeled training example. In Appendix A we show that this question can be answered, at least approximately, by computing the mutual information between label and truth.

Making Better Use of Noisy Labels for Diabetic Retinopathy Classification

We are interested in noisy datasets of medical images where many different doctors have provided labels but each image has only been labeled by a few doctors and most of the doctors have only labeled a fairly small fraction of the images. This paper focuses on datasets of images used for screening diabetic retinopathy because neural networks have recently achieved human-level performance on such images (Gulshan et al. 2016) and if we can produce even a relatively
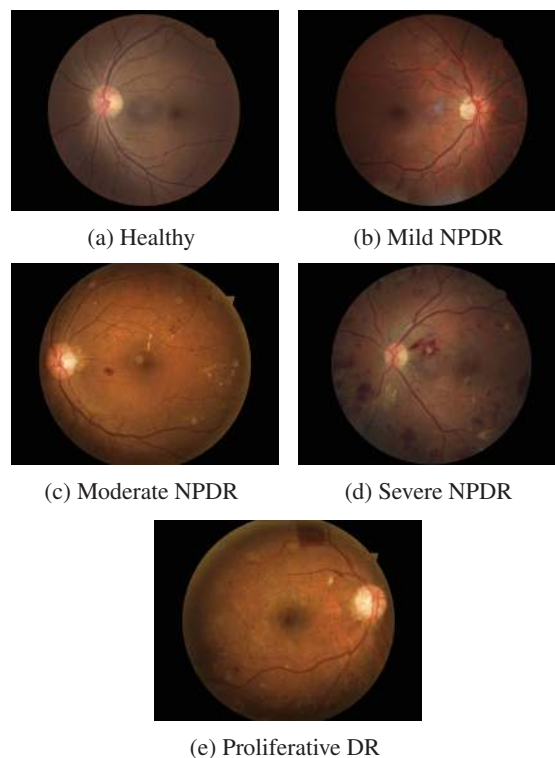
small improvement in the state-of-the-art system it will be of great value.

Diabetic retinopathy (DR) is the fastest growing cause of blindness worldwide, with nearly 415 million diabetics at risk (International Diabetes Foundation 2015). Early detection and treatment of DR can reduce the risk of blindness by 95% (National Eye Institute 2015). One of the most common ways to detect diabetic eye disease is to have a specialist examine pictures of the back of the eye called fundus images and rate them on the International Clinical Diabetic Retinopathy scale (American Academy of Ophthalmology 2002), defined based on the type and extent of lesions (e.g. microaneurysms, hemorrhages, hard exudates) present in the image. The image is classified into one of 5 categories consisting of (1) *No DR*, (2) *Mild NPDR (non-proliferative DR)*, (3) *Moderate NPDR*, (4) *Severe NPDR*, and (5) *Profilerative DR* (Figure 3). Another important clinical diagnosis that can be made from the fundus image is the presence of diabetic macular edema (DME). While this work focuses only on the 5 point grading of DR, the findings should be applicable to DME diagnosis as well.

Most of the prior work on DR classification focuses on obtaining a single ground truth diagnosis for each image, and then using that for training and evaluation. Deep learning has recently been used within this setting by Gulshan et al. (2016) who show a high sensitivity (97.5%) and specificity (93.4%) in the detection of referable DR (moderate or more severe DR).

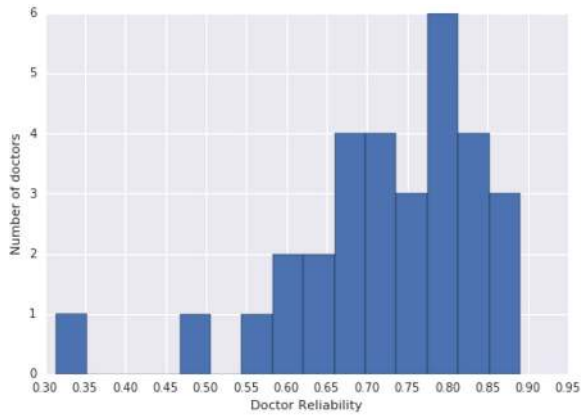In this work we explore whether, in the context of data in

Figure 4: Histogram of doctor reliabilities. These are calculated from the expectation-maximization algorithm in Welinder and Perona (2010) on our training data.

which every example is labeled by multiple experts, a better model can be trained by predicting the opinions of the individual experts as opposed to collapsing the many opinions into a single one. This allows us to keep the information contained in the assignment of experts to opinions, which should be valuable because experts labelling data differ from each other in skill and area of expertise (as is the case with our ophthalmologists, see Figure 4). Note that we still need a single opinion on the test set to be able to evaluate the models. To that end, we use a rigorous adjudicated reference standard for evaluation, where a committee of three retinal specialists resolved disagreements by discussion until a single consensus is achieved.

Related Works

Our work on learning from multiple noisy annotators relates to literature on noisy labels, crowd-sourcing, weak supervision, semi-supervised learning, item response theory, and multi-view learning.

Since the foundational work of Dawid and Skene (1979), who model annotator accuracies with expectation-maximization (EM), and Smyth et al. (1995), who integrate the opinions of many experts to infer ground truth, there has a large body of work using EM approaches to estimate accurate labels for datasets annotated by multiple experts (Whitehill et al. 2009; Raykar et al. 2009; Raykar and Yu 2012; Welinder and Perona 2010).

Works that use Bayesian probabilistic models for the image generation and/or annotation process include Welinder et al. (2010); Raykar et al. (2010); (Wauthier and Jordan 2011); Moreno et al. (2015). Yan et al. (2011); Rodrigues, Pereira, and Ribeiro (2014) learn from multiple annotators how to do active learning, i.e. which samples to select and which annotators to query, the latter using Gaussian processes to explicitly handle uncertainty. Karger, Oh, and Shah (2011); (Liu, Peng, and Ihler 2012) propose message passing algorithms for crowdsourcing.

An extension in weak supervision is generalizing from noisy sources to programmatically generate labeled training sets (Ratner et al. 2016). An extension in the crowdsourcing domain is budget allocation during label sourcing (Chen, Lin, and Zhou 2013) .

Previous work in biostatistics and epidemiology that estimate ground truth from multiple annotators in the absence of ground truth data are Rutjes et al. (2007); Hui and Walter (1980); Albarqouni et al. (2016) but none of these model individual labelers as we do.

Methods

Motivation for Model Design

First we describe the rationale behind our proposed models. There is more information in the particular labels produced by particular doctors than is captured by simply taking the average of all the doctors who have labeled a particular image and treating this distribution as the correct answer. The amount of constraint that a training case imposes on the weights of a neural network depends on the amount of information required to specify the desired output. So if we force the network to predict what each particular doctor would say for each particular training case we should be able to get better generalization to test data, provided this does not introduce too many extra parameters. For a K-way classification task, we can replace the single softmax (Goodfellow, Bengio, and Courville 2016) that is normally used by as many different K-way softmaxes as we have doctors. Of course, there will be many doctors who have not labeled a particular training image, but this is easily handled by simply not backpropagating any error from the softmaxes that are used to model those doctors. At test time we can compute the predictions of all of the modeled doctors and average them. Our belief is that forcing a neural network to model the individual doctors and then averaging at test time should give better generalization than simply training a neural network to model the average of the doctors.

We expect some doctors to be more reliable than others and we would like to give more weight to their opinions. We should this be able to do better than just averaging the opinions of the modeled doctors. After we have finished learning how to model all of the individual doctors we can learn how much to weight each modeled doctor's opinion in the averaging. This allows us to downweight the unreliable doctor models. We also expect that the doctors will have received different training and may have experienced different distributions of images so that the relative reliability of two doctors may depend on both the class of the image and on properties of the image such as the type of camera used. Our weights for averaging doctor models should therefore possibly be image-dependent.

Model Architecture

With these intuitions in mind, we consider a sequence of models of increasing complexity for training the diabetic retinopathy classifier (Figure 7). The neural network base used in this work is the Inception-v3 architecture (Szegedy et al. 2016) (Figure 6).
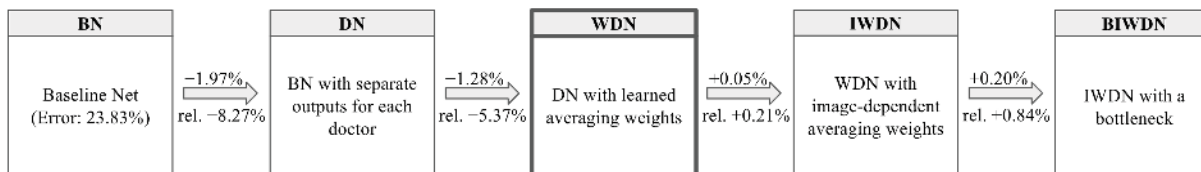
Figure 5: Description of nets used in paper. The baseline net has five-class classification error rate of 23.83% on the test dataset. The numbers above arrows refer to absolute changes in test error while the numbers below arrows refer to relative changes (negative values represent improvements). WDN (highlighted) was the optimal net.

- **Baseline Net (BN)**: Inception-v3 trained on average opinions of doctors; a TensorFlow reimplementation of the model used in Gulshan et al. (2016).

- **Doctor Net (DN)**: BN extended to model the opinions of each of the 31 doctors.

- **Weighted Doctor Net (WDN)**: Fixed DN with averaging weights for combining the predictions of the doctor models learned on top, one weight per doctor model.

- **Image-specific WDN (IWDN)**: WDN with averaging weights that are learned as a function of the image.

- **Bottlenecked IWDN (BIWDN)**: IWDN with a small bottleneck layer for learning the averaging weights.

For BN, the outputs of the last hidden layer of Inception are used to compute the logits used in the five-way softmax output layer. For DN, the opinions of each doctor are modeled using a separate softmax for each doctor, while Inception weights were shared. For evaluation, the predictions from the softmax "doctor models" are arithmetically averaged to give a single five-class prediction. For subsequent nets, the parameters and predictions of the DN model are frozen and only the averaging weights for the doctor models are learned. For WDN, one averaging weight per doctor is trained, shared across all images. For IWDN, these averaging weights are made image-dependent by letting them be a function of the last hidden layer of Inception. For BIWDN, a linear bottleneck layer of size three is added between the last hidden layer of Inception (of dimension 2048) and the 31-way softmax of IWDN as a precautionary measure against model overfitting. A bottleneck layer of this size reduces the number of trainable parameters ten-fold.

Rather than directly learning the averaging weight for each doctor model (B)(I)WDN, we learn averaging logits for each model that we could then pass through a softmax to produce averaging weights that are guaranteed to be positive. To train the averaging logits, we use the opinions of the doctors who actually labeled a training image to define the target output distribution for that image (Appendix B.2 discusses an alternative target). We then combine the predictions of the models of all the other doctors using the weights defined by their current averaging logits. Finally we update our parameters by backpropagating with the cross entropy loss between the target distribution and the weighted average prediction. This way all of the training cases that a doctor did *not* label can be used to learn the averaging logit for that doctor, and no extra data are needed beyond those used to learn the weights of DN. Moreover, if a doctor model has

Table 1: Prediction inputs to cross entropy loss for each model during training. The notation is given in the text. Note that the target is always $\frac{1}{|I|} \sum_{i \in I} l_i$.

Model	Training	Evaluation
BN	$p_\emptyset$	$p_\emptyset$
DN	$p_i, \forall i \in I$	$\frac{1}{31} \sum_{i=1}^{31} p_i$
(B)(I)WDN	$\frac{\sum_{i \notin I} p_i \cdot w_i}{\sum_{i \notin I} w_i}$	$\sum_{i=1}^{31} p_i \cdot w_i$

similar performance to other doctor models but makes very different errors it will tend to be upweighted because it will be more useful in the averaging. This upweighting of diverse doctor models would not occur if we had computed the reliabilities of the doctors separately.

For a single image, let I be the set of indices of the doctors who actually graded that image. Let the label of doctor $i \in I$ be l_i. For every doctor $j \in \{1, 2, \ldots, 31\}$, denote the prediction of its model p_j. Let $p_\emptyset$ be the prediction of the model of the average doctor in BN. For WDN, IWDN, and BIWDN, let w_j be the averaging weight for the jth modeled doctor, where $\sum_j w_j = 1$. Note that p_j is a five-dimensional vector and w_j is a scalar. The explicit inputs of the cross entropy loss being minimized during training of each model are shown in Table 1 and post-Inception computations are shown schematically in Figure 7. In the case of DN, the cross entropy losses of the individual doctor models are added together to get the total loss for each training example.

Summary of Procedure

Here we summarize the entire procedure for using Weighted Doctor Net (WDN), which turns out to be the best performing model. The process is illustrated for generic labelers in Figure 1.

WDN has two phases of training:

- **Phase 1**: We learn a doctor model for each doctor. Each doctor model consists of the Inception-v3 base followed by a softmax output layer. The Inception-v3 is shared by all the doctor models while the output layers are unique to each doctor model.

- **Phase 2**: We fix the doctor models that we learned in Phase 1 (Note this implies that the predictions made by the doctor models for any given image are also fixed.)

Now we learn how to combine the predictions of the doctor models in a weighted manner. We do this by training averaging logits according to Table 1 and then taking a softmax of these averaging logits to get averaging weights.

During evaluation of WDN, the prediction made for our model is a linear combination of the doctor models predictions where the coefficients are the averaging weights learned in Phase 2 of training.

Next we describe two benchmarks to compare our models against.

Estimating Doctor Reliability with EM

Welinder and Perona (2010) use a representative online EM algorithm to estimate abilities of multiple noisy annotators and to determine the most likely value of labels. We calculate updated labels by executing the method in Welinder and Perona (2010) on our human doctors and we use these updated labels to train BN, as a competing algorithm for our DN method. Welinder and Perona (2010) also actively select which images to label and how many labels to request based on the uncertainty of their estimated ground truth values and the desired level of confidence, and they select and prioritize which annotators to use when requesting labels. We do not use these other aspects of their algorithm because labels for all images in our dataset have already been collected.

Modeling Label Noise

Mnih and Hinton (2012) describe a deep neural network that learns to label road pixels in aerial images. The target labels are derived from road maps that represent roads using vectors. These vectors are converted to road pixels by using knowledge of the approximate width of the roads so the target labels are unreliable. To handle this label noise, Mnih and Hinton (2012) propose a robust loss function that models asymmetric omission noise.

They assume that a true, unobserved label $\mathbf{m}$ is first generated from a $w_m \times w_m$ image patch $\mathbf{s}$ according to some distribution $p(\mathbf{m}|\mathbf{s})$, and the corrupted, observed label $\tilde{\mathbf{m}}$ is then generated from $\mathbf{m}$ according to a noise distribution $p(\tilde{\mathbf{m}}|\mathbf{m})$. The authors assume an asymmetric binary noise distribution $p(\tilde{m}_i|m_i)$ that is the same for all pixels i. They assume that conditioned on $\mathbf{m}$, all components of $\tilde{\mathbf{m}}$ are independent and that each $\tilde{m}_i$ is independent of all $m_{j\neq i}$. The observed label distribution is then modeled as:

$$p(\tilde{\mathbf{m}}|\mathbf{s}) = \prod_{i=1}^{w_m^2} \sum_{m_i} p(\tilde{m}_i|m_i)p(m_i|\mathbf{s}).$$

For another baseline, we use a multi-class extension of their method on DN, modeling the noise distribution prior for all doctors d with the parameters:

$$\theta_{ll'} = p(\tilde{m}_d = l'|m_d = l)$$

where $l, l' \in \{1, 2, 3, 4, 5\}$. We estimate $\theta_{ll'}$ using the 5×5 confusion matrix between individual and average doctor opinions on training images. Treating the average doctor opinion as the true label, we convert each doctor's individual count matrix into proportions which we averaged

across all doctors. We train this model by minimizing the negative log posterior, $-\log(p(\tilde{\mathbf{m}}|\mathbf{s}))$. This variant of the method by Mnih and Hinton (2012) is an alternative way to improve upon DN to our proposal of learning averaging weights (WDN).

Experimental Setup

Neural Network Training

We train the network weights using distributed stochastic gradient descent (Abadi et al. 2016) with the Adam optimizer on mini-batches of size 8. We train using TensorFlow with 32 replicas and 17 parameter servers, with one GPU per replica. To speed up the training, we use batch normalization (Ioffe and Szegedy 2015), pre-initialization of our Inception network using weights from the network trained to classify objects in the ImageNet dataset (Russakovsky et al. 2015), and the following trick: we set the learning rate on the weight matrix producing prediction logits to one-tenth of the learning rate for the other weights. We prevent overfitting using a combination of L1 and L2 penalties, dropout, and a confidence penalty (Pereyra et al. 2017), which penalizes output distributions with low entropy. At the end of training, we use an exponentially decaying average of the recent parameters in the final model.

We tune hyperparameters and pick model checkpoints for early stopping on the validation dataset, using five-class classification error rate as the evaluation metric. The optimal values for these hyperparameters are displayed in Appendix C. Note that we tune the baseline as well to ensure that our improvements are not the result of more hyperparameter optimization. When evaluating on the test set we average the predictions for the horizontally and vertically flipped versions (four in total) of every image.

We also train a version of BN where the output prediction is binary instead of multi-class, as is done in Gulshan et al. (2016). The binary output is obtained by thresholding the five-class output at the *Moderate NPDR* or above level, a commonly used threshold in clinics to define a referable eye condition. For this BN-binary network, the area under the ROC curve is used as the validation evaluation metric.

To deal with differences in class distribution between the datasets (Table 2), we use log prior correction during evaluation. This entails adding to the prediction logits, for each class, the log of the ratio of the proportion of labels in that class in the evaluation dataset to the proportion of labels in that class in the training set. Our assumed test class distribution for computing the log prior correction is the mean distribution of all known images (those of the training and validation sets). So for each image under evaluation we update the prediction logit for class c by adding:

$$\log\left(\frac{q_{valid}(c)}{q_{train}(c)}\right) \qquad \text{for the validation dataset, and}$$

$$\log\left(\frac{q_{valid\cup train}(c)}{q_{train}(c)}\right) \qquad \text{for the test dataset,}$$

where $q(c)$ is the proportion of labels in that class. Applying log prior correction improves accuracy and all our reported results use it.

Datasets

The training dataset consists of $126,522$ images sourced from patients presenting for diabetic retinopathy screening at sites managed by 4 different clinical partners: EyePACS, Aravind Eye Care, Sankara Nethralaya, and Narayana Nethralaya. The validation dataset consists of 7,804 images obtained from EyePACS clinics. Our test dataset consists of 3,547 images from the EyePACS-1 and Messidor-2 datasets. More details on image sourcing are in Appendix D.

Each of the images in the training and validation datasets was graded by at least one of 54 US-licensed ophthalmologist or ophthalmology trainee in their last year of residency (postgraduate year 4). For training the doctor models, we use the 30 ophthalmologists who graded at least 1,000 images were used, and we lump the remaining doctors as a single composite doctor to avoid introducing doctor-specific parameters that are constrained by less than 1,000 training cases. Meanwhile, the labels for the test set were obtained through an adjudication process: three retina specialists graded all images in the test dataset, and discussed any disagreements as a committee until consensus was reached.

We scale-normalize our images by detecting the circular fundus disk and removing the black borders around them. We use images at a resolution of $587{\times}587$ pixels and we augment our training data with random perturbations to image brightness, saturation, hue, and contrast.

Our Baseline vs Published Baseline

This section describes the multiple ways in which our baseline differs from that of Gulshan et al. (2016). For these reasons, results from this paper's own BN should be used for model comparisons with DN, WN, IWDN, and BIWDN rather than numbers from Gulshan et al. (2016).

- Unlike in Gulshan et al. (2016), we remove grades of doctors who grade test set images from both training and validation sets to reduce the chance that the model is overfitting on certain experts. This handicaps our performance vis-à-vis their paper, especially because we exclude the most expert doctors (the retinal specialists) during model development, but ensures generalizability of our results.

- We use different datasets, and in particular our adjudicated test set has gold standard "ground truth" labels.

- We train with five-class loss instead of binary loss.

- If a doctor grades a single image multiple times, as often occurs, Gulshan et al. (2016) treats these as independent diagnoses while we collapse these multiple diagnoses into a distribution over classes.

- We employ higher resolution images ($587{\times}587$ pixels versus $299{\times}299$) and image preprocessing and theoretical techniques unused in Gulshan et al. (2016).

Summary of Results

We run 10 replicates of each model and average the resulting metrics, which are reported in Table 4. For full comparability of models we use the same 10 replicates reported for DN to serve as the fixed part of the model for training the WDN, IWDN, and BIWDN replicates.

Table 2: Class distributions of datasets (as %).

Grade	Training	Validation
1	51.03	72.69
2	24.75	17.62
3	16.81	7.27
4	4.17	1.20
5	3.23	1.21

Table 3: Test metrics from training with Multi-class vs Binary loss for BN.

Test Metric (%)	Binary loss	Five-class loss
Binary AUC	95.58	97.11
Binary Error	11.27	9.92
Spec@97%Sens	63.12	79.60

Training with Five-Class Loss Beats Training with Binary Loss Even on Binary Metrics

We find that training BN with a five-class loss improves test binary AUC compared to training with a binary loss, as is done by Gulshan et al. (2016), even when validating the former on five-class training error instead of binary AUC (Table 3). Test binary AUC is raised by a substantial 1.53% (97.11% vs 95.58%) from using five-class loss. Intuitively this fits with our thesis that generalization is improved by increasing the amount of information in the desired outputs. All results reported in Table 4 and subsequent sections, including for BN, are obtained from training with five-class loss.

Averaging Modeled Doctors Beats Modeling the Average Doctor

We see a reduction in five-class classification test error of 1.97% (from 23.83% to 21.86%) from averaging modeled doctors (DN) instead of modeling the averaged doctor (BN). In comparison, using labels from the algorithm in Welinder and Perona (2010) to train BN only reduces five-class test classification error by 0.09%. Over BN, DN also increases binary AUC by 0.17% (97.11% to 97.28%), decreases binary classification error 0.17% (9.92% to 9.75%), and increases specificity at 97% sensitivity (spec@97%sens) by 2.21% (79.60% to 81.81%). Meanwhile, using labels from Welinder and Perona (2010) on BN merely increases spec@97%sens by 0.37% relative to vanilla BN and actually leads to slightly worse performance on binary AUC (-0.11%) and binary error (+0.20%). Note that the binary AUC, binary error, and spec@97%sens metrics would be improved for all models if we were to do hyperparameter tuning and early stopping for them specifically, but we decided to do all our model selection on one metric (five-class error) both for simplicity and to simulate the decision metric required in real-life automated diagnosis systems. We see that DN is significantly better on all test metrics compared to BN trained using the labels obtained from the algorithm in Welinder and Perona (2010).
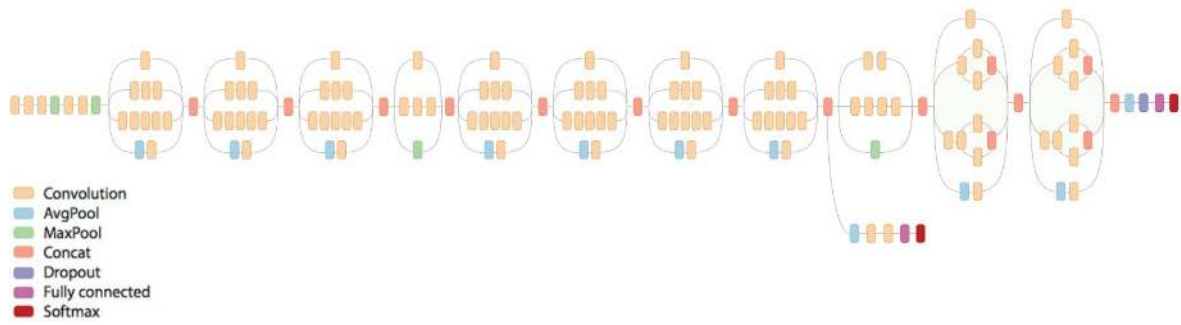
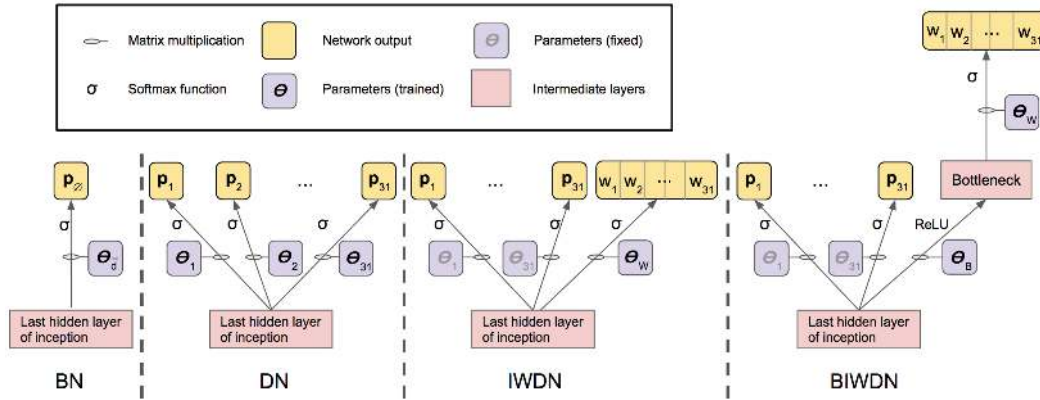Figure 6: Schematic diagram of Inception-v3 (Shlens 2016).

Figure 7: Schematic diagram of nets. These schematics show how the parameters, network outputs, and averaging weights for doctor models are connected. Table 1 lists how the outputs are used in a loss function for training. In WDN (not shown in figure), the averaging logits are not connected to the last hidden layer of Inception and are just initialized from a constant vector.

Learning Averaging Weights Helps

We see a further 1.28% decrease in five-class test error from using WDN as opposed to DN. Binary AUC increases an additional 0.17%, binary classification error decreases another 0.68%, and spec@97%sens increases an extra 0.88%, all on test data. Results from IWDN and BIWDN are slightly worse than those from WDN. We would expect a bigger improvement from WDN and potentially further improvements from training averaging logits in an image-specific way if we had doctors with more varied abilities and greater environmental differences, but on our dataset image-specific averaging logits does not help. Our extension of the competing algorithm by Mnih and Hinton (2012) actually causes DN to perform worse by 0.90% on five-class classification test error, and is more computationally costly than (B)(I)WDN. A different noise model we considered does not help either (Appendix B.3).

Conclusion

We introduce a method to make more effective use of noisy labels when every example is labeled by a subset of a larger pool of experts. Our method learns from the identity of multiple noisy annotators by modeling them individually with a shared neural net that has separate sets of outputs for each expert, and then learning averaging weights for combining their modeled predictions. We evaluate our method on the diagnosis of diabetic retinopathy severity on the five-point scale from images of the retina. Compared to our baseline model of training on the average doctor opinion, a strategy that yielded state-of-the-art results on automated diagnosis of DR, our method can lower five-class classification test error from 23.83% to 20.58%. We also find that, on binary metrics, training with a five-class loss significantly beats training with a binary loss, as is done in the published baseline. We compare our method to competing algorithms by Welinder and Perona (2010); Mnih and Hinton (2012) and we show that corresponding parts of our method give superior performance to both. Our methodology is generally applicable to supervised training systems using datasets with labels from multiple annotators.

Table 4: Summary of Test Results. All models in this table are trained with five-class loss except DN Mnih, whose loss was the negative log posterior.

Metric (%)	BN		DN		WDN	IWDN	BIWDN
	$\emptyset$	Welinder	$\emptyset$	Mnih			
five-class Error	23.83	23.74	21.86	22.76	**20.58**	20.63	20.83
Binary AUC	97.11	97.00	97.28	97.42	**97.45**	97.43	97.41
Binary Error	9.92	10.12	9.75	10.24	**9.07**	9.12	9.23
Spec@97%Sens	79.60	79.97	81.81	**83.61**	82.69	82.46	82.46

Table 5: Optimal Hyperparameters from Grid Search. Note that the learning rate for doctor models is one-tenth the learning rate for the rest of the network listed here. WD=weight decay, wel=welinder

Hyperparameter	BN binary	BN	BN wel	DN	DN mnih	WDN	IWDN	BIWDN
Learning rate	0.0001	0.0003	0.0003	0.001	0.0003	0.03	1×10^{-6}	3×10^{-7}
Dropout for Inception	0.75	0.95	0.95	0.85	0.95	-	-	-
Dropout for output heads	0.8	0.85	0.85	0.9	0.9	-	-	-
Entropy weight	0.0125	0.025	0.015	0.0175	0.02	0.0225	0.005	0.0125
L2 WD for Inception	0.01	0.01	0.01	0.001	0.004	-	-	-
L1 WD for doctor models	0.001	0.00004	0.0001	0.001	0.01	-	-	-
L2 WD for doctor models	0.01	0.004	0.001	0.01	0.04	-	-	-
L1 WD for averaging logits	-	-	-	-	-	0.4	0.02	4
L2 WD for averaging logits	-	-	-	-	-	15	0.4	110
Bottleneck size	-	-	-	-	-	-	-	3

A. Mutual Information for Noisy Labels

Here we compute the mutual information between a noisy MNIST label and the truth, assuming random noise, in order to estimate the number of noisily labeled training cases equivalent to one case that is known to be correctly labeled.

Empirically, N perfectly labeled training cases give about the same test error as $N I_{\text{perfect}}/I_{\text{noisy}}$ training cases with noisy labels, where I_{noisy} is the mutual information per case between a noisy label and the truth and I_{perfect} is the corresponding mutual information for perfect labels. For ten classes, the mutual information (in nats) is $I_{\text{perfect}} = 2.3 = -\log(0.1)$, but when a noisy label is 20% correct on average, the mutual information is:

$$I_{\text{noisy}} = 0.044$$

$$= -\log(0.1) - 10 \times 0.02 \times \log\left(\frac{0.1}{0.02}\right)$$

$$- 90 \times 0.1 \times \frac{0.8}{9} \log\left(\frac{0.1}{0.1 \times 0.8/9}\right).$$

So if the learning is making good use of the mutual information in the noisy labels we can predict that 60,000 noisy labels are worth $60{,}000 \times 0.044/2.3 \approx 1{,}148$ clean labels. In reality we needed about 1,000 clean labels to get similar results.

B. Other Ideas Tested

B.1 Mean Class Balancing

In addition to log prior correction of class distributions, we also attempted mean class balancing wherein samples from less frequent classes were upweighted and more frequent classes are downweighted in the cross entropy loss, in inverse proportion to their prevalence relative to the uniform distribution across classes. Explicitly, we weight each sample of class c by:

$$\alpha_c = \frac{\bar{q}}{q(c)} = \frac{1}{|c|q(c)}.$$

Eigen and Fergus. (2015) employ a similar method for computer vision tasks although they use medians instead of means. In our case, using mean class balancing lowers performance, possibly because it makes too many assumptions on the unknown test distribution, and was not employed.

B.2. Alternative Target Distribution for Training Averaging Logits

To train the averaging logits, we take each training case and use the opinions of the doctors who actually labeled that case to define the target output distribution. Alternatively, the target distribution can be defined as the equally weighted average of the predictions of the doctor models corresponding to the doctors who labeled that case. In the notation used in Table 1, this would be $\frac{1}{|I|}\sum_{i\in I} p_i$. We experimented with using this alternative target distribution in calculating cross entropy loss but saw inferior results.

B.3. An Alternative Noise Model

Because our multi-class extension of Mnih and Hinton (2012) shows poor results, which we postulated may have been because it is sensitive to differences in class distributions between datasets, we considered a different noise model that makes less assumptions on the class distribution

of the data. This model assumes a symmetric noise distribution that is determined by a single prior parameter. This assumes that if a label is wrong, it has equal probability of belonging to any of the other classes. However we allow this parameter to vary by doctor and we estimate it for each doctor d as

$$\theta_d = p(\tilde{m}_d = l | m_d = l),$$

where the real doctor reliability score is calculated from the Welinder and Perona (2010) algorithm. Unfortunately this method performs slightly worse than the 5-class variant of Mnih and Hinton (2012). Note that a number of other noise models of varying complexity can be considered as well.

C. Hyperparameter Search

Table 5 displays the optimal hyperparameters used in DR classification. We tuned using grid search on the following hyperparameter spaces: dropout for Inception backbone $\in \{0.5, 0.55, 0.6, \ldots, 1.0\}$, dropout for doctor models $\in \{0.5, 0.55, 0.6, \ldots, 1.0\}$, learning rate $\in \{1 \times 10^{-7}, 3 \times 10^{-7}, 1 \times 10^{-6}, \ldots, 0.03\}$, entropy weight $\in \{0.0, 0.0025, 0.005, \ldots, 0.03\} \cup \{0.1\}$, weight decay for Inception $\in \{0.000004, 0.00001, 0.00004, \ldots, 0.1\}$, L1 weight decay for doctor models $\in \{0.000004, 0.00001, 0.00004, \ldots, 0.04\}$, L2 weight decay for doctor models $\in \{0.00001, 0.00004, \ldots, 0.04\}$, L1 weight decay for averaging logits $\in \{0.001, 0.01, 0.02, 0.03, \ldots, 0.1, 0.2, 0.3, \ldots, 1, 2, 3, \ldots, 10, 100, 1000\}$, L2 weight decay for averaging logits $\in \{0.001, 0.01, 0.1, 0.2, 0.3, \ldots, 1, 5, 10, 15, 20, 30, \ldots, 150, 200, 300, 400, 500, 1000\}$, and bottleneck size (for BIWDN) $\in \{2, 3, 4, 5, 6, 7\}$. We used a learning rate decay factor of 0.99 optimized for BN. The magnitudes of the image preprocessing perturbations were also optimized for BN.

D. Dataset Details

Our training set consists of 119,589 of the 128,175 images used in the training set of Gulshan et al. (2016) and 6,933 new labeled images acquired since the creation of their training dataset. The images in the training set of Gulshan et al. (2016) that we do not use were excluded for the following reasons. (i) 4,204 images of their dataset were removed to create a separate validation dataset for experiments within the research group, (ii) 4,265 were excluded because they were deemed ungradable by every ophthalmologist that graded them. Unlike Gulshan et al. (2016), we do not predict image gradeability in this work and hence excluded those images. (iii) 117 of their images were excluded because they fail our image scale normalization preprocessing step.

Our validation dataset consists of 7,963 images obtained from EyePACS clinics. These images are a random subset of the 9,963 images of the EyePACS-1 test set used in Gulshan et al. (2016). The remaining 2,000 images were used as part of our test set. In practice, only 7,805 of the 7,963 validation images have at least one label, since the remaining 158 images were of poor quality and considered ungradable by all ophthalmologists that labeled them.

The test set consists of 1,748 images of the Messidor-2 dataset (Decencière et al. 2014) and 2,000 images of the EyePACS-1 test dataset used in Gulshan et al. (2016), as we just mentioned. 1,744 of the 1,748 images of Messidor-2 and 1,803 of the 2,000 images from EyePACS-1 were considered gradable after adjudication and were assigned labels.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; and Zheng, X. 2016. Tensorflow: A system for large-scale machine learning. *OSDI*.

Albarqouni, S.; Baur, C.; Achilles, F.; Belagiannis, V.; Demirci, S.; and Navab, N. 2016. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging* 35(5):1313–1321.

American Academy of Ophthalmology. 2002. International clinical diabetic retinopathy disease severity scale. Detailed table. icoph.org/dynamic/attachments/resources/diabetic-retinopathy-detail.pdf.

Chen, X.; Lin, Q.; and Zhou, D. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *ICML*, 64–72.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics* 28(1):20–28.

Decencière, E.; Xiwei, Z.; Cazuguel, G.; Lay, B.; Cochener, B.; Trone, C.; Gain, P.; nez Varela, J.-R. O.; Massin, P.; Erginay, A.; Charton, B.; and Kleain, J.-C. 2014. Feedback on a publicly distributed image database: The messidor database. *Image Anal Stereol* 33(3):231–234.

Eigen, D., and Fergus., R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV* 11(18):2650–2658.

Elmore, J. G.; Wells, C. K.; Lee, C. H.; Howard, D. H.; and Feinstein, A. R. 1994. Variability in radiologists interpretations of mammograms. *NEJM* 331(22):1493–1499.

Elmore, J. G.; Longton, G. M.; Carney, P. A.; Geller, B. M.; Onega, T.; Tosteson, A. N. A.; Nelson, H. D.; Pepe, M. S.; Allison, K. H.; Schnitt, S. J.; O'Malley, F. P.; and Weaver, D. L. 2015. Diagnostic concordance among pathologists interpreting breast biopsy specimens. *JAMA* 313(11):1122–1132.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. Cambridge, Massachusetts: MIT Press.

Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M. C.; D. Wu, A. N.; Venugopalan, S.; Widner, K.; T; Madams; Cuadros, J.; Kim, R.; Raman, R.; Nelson, P. C.; Mega, J. L.; and Webster, D. R. 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA* 316(22):2402–2410.

Hui, S. L., and Walter, S. D. 1980. Estimating the error rates of diagnostic tests. *Biometrics* 167–171.

International Diabetes Foundation. 2015. Idf diabetes atlas, 7th ediction. diabetesatlas.org.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML* 37:448–45.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. *NIPS* 24:1953–1961.

Kingma, D. P., and Ba, J. L. 2015. Adam: A method for stochastic optimization. *ICLR*.

Liu, Q.; Peng, J.; and Ihler, A. T. 2012. Variational inference for crowdsourcing. *NIPS* 25:692–700.

Mnih, V., and Hinton, G. E. 2012. Learning to label aerial images from noisy data. *ICML* 567–574.

Moreno, P. G.; Artés-Rodríguez, A.; Teh, Y. W.; and Perez-Cruz, F. 2015. Bayesian nonparametric crowdsourcing. *JMLR*.

National Eye Institute. 2015. Facts about diabetic eye disease. nei.nih.gov/health/diabetic/retinopathy.

Pereyra, G.; Tucker, G.; Kaiser, L.; and Hinton, G. E. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv*.

Ratner, A. J.; De Sa, C. M.; Wu, S.; Selsam, D.; and Ré, C. 2016. Data programming: Creating large training sets, quickly. In *NIPS*, 3567–3575.

Raykar, V. C., and Yu, S. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *JMLR* 13:491–518.

Raykar, V.; Yu, S.; Zhao, L.; Jerebko, A.; Florin, C.; Valadez, G.; Bogoni, L.; and Moy, L. 2009. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. *ICML* 26:889–896.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *JMLR* 11:1297–1322.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Gaussian process classification and active learning with multiple annotators. In *ICML*, 433–441.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. Imagenet large scale visual recognition challenge. *JMLR* 115(3):211–252.

Rutjes, A.; Reitsma, J.; Coomarasamy, A.; Khan, K.; and Bossuyt, P. 2007. Evaluation of diagnostic tests when there is no gold standard. a review of methods. *Health Technology Assessment* 11.

Shlens, J. 2016. Train your own image classifier with inception in tensorflow. research.googleblog.com/2016/03/train-your-own-image-classifier-with.html.

Smyth, P.; Fayyad, U.; Burl, M.; Perona, P.; and Baldi, P. 1995. Inferring ground truth from subjective labelling of venus images. *NIPS* 7:1085–92.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Re-thinking the inception architecture for computer vision. *CVPR* 2818–2826.

Wauthier, F. L., and Jordan, M. I. 2011. Bayesian bias mitigation for crowdsourcing. *NIPS* 24:1800–1808.

Welinder, P., and Perona, P. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. *CVPR Workshop* 25–32.

Welinder, P.; Branson, S.; Perona, P.; and Belongie, S. J. 2010. The multidimensional wisdom of crowds. In *NIPS*, 2424–2432.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *NIPS* 22:2035–2043.

Yan, Y.; Fung, G. M.; Rosales, R.; and Dy, J. G. 2011. Active learning from crowds. In *ICML*, 1161–1168.