# Confidence measures for CNN classification using Gaussian processes

**Charlie Hewitt** [1]

## Abstract

This paper presents a hybrid classification technique using Gaussian processes (GP) fitted on features extracted by a convolutional neural network (CNN) to enable estimation of prediction confidence. The hybrid classifier is shown to outperform the base CNN for the MNIST dataset. The variance values predicted by the GP are demonstrated to be useful in estimating significance of classifications and confidence intervals. A similar approach is evaluated for a simple regression task and shown to be markedly less effective. Finally, the implications of such confidence measures are discussed for real-world applications.

## 1. Introduction

### 1.1. Motivation

Convolutional neural networks (CNN) provide state-of-the-art performance for a large number of image-based machine learning (ML) tasks such as segmentation, classification and regression. CNNs typically achieve very high accuracy, but give no real indication of confidence in their predictions.

CNNs are particularly popular for medical applications and have already been applied to a number of medical imaging problems (Litjens et al.). For example, detecting brain lesions (Kamnitsas et al.) and volumetric segmentation for MRI scans (Milletari et al., 2016), as well as for the task of drug discovery (Wallach et al., 2015).

For medical contexts in particular, it is critical to have a measure of confidence in a prediction, as an error may be life threatening. At present the CNN models used do not provide this measure of confidence, a fact which is rightly reducing the willingness to adopt these new technologies within medicine. There is clearly great potential for improved patient care through the use of ML technologies, and confidence in the predictions of these models is a key factor in enabling this.

While medicine is one of the most obvious fields where confidence is critical, there are of course many other challenges to which CNNs have been applied where confidence is important. One example being the many vision related tasks involved in autonomous driving, such as vehicle and sign detection (Li et al., 2016; Sermanet & LeCun, 2011), as well as semantic segmentation (Cordts et al., 2016; Garcia-Garcia et al., 2017). For almost all applications a measure of confidence is at least useful, for deep neural networks (DNN) in general, not just for CNNs.

Consequently, the concept of augmenting a conventional CNN with an ML model which provides confidence measures is considered. The aim being to exploit the performance of CNN models for image-based tasks, while also gaining a indication of prediction confidence. Gaussian Processes (Rasmussen, 2004) (GP) are an obvious candidate to provide this confidence measure, due to their mathematically rigorous derivation and consequent ability to predict meaningful variances. One key benefit is that GPs do not extrapolate prediction confidence, so a new data point which is far from the observed data should be indicated by a low prediction confidence, unlike for CNNs which typically construct a hard decision boundary.

### 1.2. Contributions & Paper Structure

The contributions of this paper are threefold:

- Evaluation of classification confidence estimation methodologies using hybrid CNN/GP classifier on the MNIST dataset and the effects of noise and adversarial perturbations on these confidence metrics.
- Investigation into confidence measures for regression tasks using a hybrid CNN/GP regression model.
- Discussion of the implications of the proposed confidence estimation techniques for real-world problems.

The remainder of Sec. 1 includes a description of the previous work related to confidence in CNN and DNN predictions. A summary of the hybrid classifier implementation is given in Sec. 2 and Sec. 3 includes the results for the system when evaluated on a number of variants of the MNIST dataset. A discussion of the implications is given in Sec. 4, along with suggestions for future work and a brief conclusion.

[1] University of Cambridge, UK. Correspondence to: Charlie Hewitt <ac@chewitt.me>.

### 1.3. Related Work

Confidence in ML models is typically represented as a prediction interval (PI), that is the range in which a value is predicted to fall with some confidence, typically 95%. A number of methods for estimating these PIs for neural networks (NN) have been proposed (Khosravi et al., 2011), many originating from the mid 1990s, though surprisingly little recent development has occurred.

The bootstrap method (Efron, 1992; Heskes, 1997) is the most commonly used method for estimating PIs. An ensemble of NNs are trained on varying subsets of the training set, the output is then taken as the mean of the predictions from each NN, and the variance calculated directly from these predictions. The bootstrap method generates relatively poor quality PIs, typically overestimating variance, but with highly reproducible results. It is quite computationally efficient depending on the number of NNs used.

The Bayesian approach (Bishop, 1995; MacKay, 1992) involves modelling NN parameters as a set of random variables with some prior distributions, and the output of each layer as having some posterior distribution. From this, the output distribution can be calculated, with the variance estimated by approximating the Hessian matrix of the cost function. The resulting PIs are of generally good quality and are highly reproducible, though this method is very inefficient for large datasets and networks. There has been some more recent work devising efficient Bayesian CNNs (Gal & Ghahramani, 2015) which may improve the feasibility of this approach.

The delta method (Hwang & Ding, 1997; De Vleaux et al., 1998) interprets neural networks as a non-linear regression model, this allows the application of asymptotic theories to construct PI. The PIs generated using this method are generally of high quality, but repeatability is poor and in some cases the resulting PIs are inaccurate. This method is also computationally intensive which is a particular problem for large networks like CNNs.

Mean value expectation (MVE) (Nix & Weigend, 1994) makes use of two separate NNs to predict the mean and variance of the output distribution independently, this can then be used to calculate PIs. The mean network can be trained ordinarily, but as the variance is not know for each training sample, the variance network is trained using a maximum likelihood estimation approach. As only two networks are required, MVE is significantly more computationally efficient than other methods, but the quality and repeatability of the PIs generated are poor and unreliable for real-world applications.

Most of these methods are not realistically applicable to the large CNNs and DNNs typically used today, or produce very poor quality PIs that serve little use in real-world scenarios. As such, alternative methods have been proposed, such as the hybrid of NNs and GPs featured here. This method is typically quite efficient as only a single NN model needs to be trained, followed by the fitting of a GP, typically in a relatively small feature space outputted from and intermediate layer of the NN.

GPDNN (Bradshaw et al., 2017) is the primary work in this relatively unexplored area, focussing on comparison between a conventional CNN and a hybrid GPDNN (a CNN with soft-max layer replaced by a GP, trained end-to-end). The authors evaluate the relative performance of these models on standard datasets and the impacts of adversarial examples in great detail. They do not, however, provide much in terms of discussion of the implications of the PIs that GPs inherently provide.

## 2. Implementation

### 2.1. CNN

A basic CNN is used for the purposes of this investigation, one typical of designs used for classification on the MNIST dataset (Lecun et al., 1998). Two convolution layers (32 and 64 deep with 3 × 3 kernels) are followed by a single max-pooling layer, a single fully connected layer with 128 nodes and finally a 10 node, fully connected layer with soft-max activation to output classification results. Dropout is used to prevent over-fitting and ReLU activation (Nair & Hinton, 2010) for all layers except the final classification output.

The CNN is implemented using Keras (Chollet et al., 2015) and trained over 32 epochs with a batch size of 128 using the Adam optimiser (Kingma & Ba, 2014) with default parameters. To generate features for use by the GP, the output layer is removed from the model, leaving a 128 dimensional output vector from the previous fully connected layer.

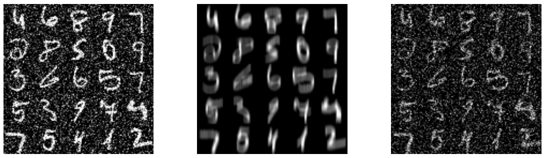
### 2.2. GP Classification

GPFlow (Matthews et al., 2017) is used to train and evaluate all GPs used for this investigation. Due to the large volume of data (60000 training examples), two implementations of scalable GP classifiers are employed.

Sparse variational GP (SVGP) (Hensman et al., 2015b) scale the standard GP model within a variational inducing point framework. This drastically reduces training time and memory requirements, as well as improving results for very large datasets.

Sparse variational GP using MCMC (SGPMC) (Hensman et al., 2015a) instead approximate both the function values and covariance parameters simultaneously using a Markov chain Monte Carlo sampling scheme. This allows for repre-

*Figure 1.* n-MNIST example images from http://csc.lsu.edu/~saikat/n-mnist/.



(a) AWGN (b) Motion blur (c) Reduced contrast and AWGN

sentation of non-Gaussian posterior distributions and therefore theoretically superior models, as well as enabling use of very large datasets.

In both cases hyper-parameters are determined automatically using GPFlow's optimisation procedure. Ten latent GPs are used for MNIST classification and for both implementations every 50th item in the training set is used to provide the inducing points. This gives a good trade-off between performance and training time.

### 2.3. Classification Datasets

For classification, the standard MNIST dataset (Lecun et al., 1998), containing 60000 $28 \times 28$ pixel, black and white training images, and 10000 similar test images, of handwritten digits from 0 to 9 is used.

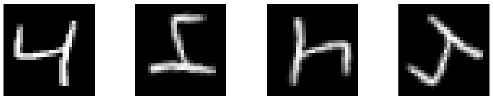
In addition to this, the noisy MNIST (n-MNIST) dataset (Basu et al., 2015) is used to asses the resilience of the classifier to a number of input image distortions. Examples from the n-MNIST dataset are shown in Fig. 1; variants include additive white Gaussian noise (AWGN), motion blur and reduced contrast with AWGN.

Each of these variants contains the same images as the original dataset (60000 training, 10000 test) with the described distortions applied. The CNN and GP models, though, are not retrained using the n-MNIST training sets. Both models are trained on MNIST and evaluated on all four test sets without retraining.

### 2.4. GP Regression

In order to also evaluate the proposed technique for regression tasks, a variant of the MNIST dataset is generated. For each image containing a number 4 in the original dataset, 10 variants are generated, rotated by a random angle between -180 and 180 degrees. Examples of such images can be seen in Fig. 2. The resulting dataset (referred to as r-MNIST) contains 58420 training images and 9820 test images. The task is then to determine the angle by which each image has been rotated; the number 4 was chosen due to its lack of rotational symmetry.

*Figure 2.* Three example r-MNIST rotated images generated for a single original MNIST image.



Orig. −83° −175° 62°

For this purpose a sparse Gaussian process regression (SGPR) (T, 2009) model is used, which utilises a similar variational formulation for sparse approximations to SVGP. Again every 50th item in the training set is used to provide the inducing points.

## 3. Results

### 3.1. Kernel Selection

A number of different GP kernels can be used, these are evaluated with an SVGP model for the MNIST dataset as well as the three n-MNIST variants. Results are shown in Table 1; the polynomial kernel outperforms all others for all datasets apart from n-MNIST-C for which the linear kernel achieves the highest accuracy. The RBF, Matern32 and Mater52 kernels also provide very poor variance predictions which are not useful for confidence estimation. As such, polynomial kernels are used for the rest of the investigation.

All results are for isotropic kernels; GPFlow provides an ARD option to enable different length scales for each input dimension, though this negatively impacted performance. Adding a white Gaussian noise kernel also had no positive effect on performance, perhaps surprising given the noise introduced to the n-MNIST input images.

A number of multi-kernel models were also investigated, none of which improved upon the performance of the standard polynomial kernel. GPFlow ordinarily shares the same kernel between all latent processes, an alternative model using a separate kernel for each latent process was evaluated, but did not improve performance. A variety of summed kernels were also investigated, though again had no positive impact.

### 3.2. Model Selection

Comparative results for the two available GP implementations, SVGP and SGPMC, are given in Table 2 along with the CNN baseline performance. SVGP has the best performance, broadly equalling the CNN baseline, and in some cases slightly outperforming the original soft-max classifier.

SGPMC seems to fit the noiseless data better (achieving significantly higher accuracy on MNIST and n-MNIST-B),

*Table 1.* Performance for various kernels on the MNIST dataset and n-MNIST variants.

| | Linear | RBF | Poly | Matern12 | Matern32 | Matern52 |
|---|---|---|---|---|---|---|
| MNIST | 0.9890 | 0.9803 | **0.9905** | 0.9894 | 0.9539 | 0.9669 |
| n-MNIST-A | 0.9324 | 0.8609 | **0.9351** | 0.9320 | 0.6847 | 0.7640 |
| n-MNIST-B | 0.9495 | 0.8735 | **0.9592** | 0.9550 | 0.7786 | 0.8240 |
| n-MNIST-C | **0.7339** | 0.4821 | 0.7222 | 0.6811 | 0.1910 | 0.2535 |

*Table 2.* SVGP and SGPMC results compared with CNN baseline for the MNIST dataset and n-MNIST variants.

| | CNN | SVGP | SGPMC | SGPMC* |
|---|---|---|---|---|
| MNIST | 0.9906 | 0.9905 | **0.9916** | 0.9912 |
| n-MNIST-A | 0.9293 | **0.9351** | 0.8762 | 0.9052 |
| n-MNIST-B | 0.9588 | 0.9592 | **0.9771** | 0.9640 |
| n-MNIST-C | 0.7200 | **0.7222** | 0.5973 | 0.6948 |

* With added white noise kernel.

*Table 3.* Percentage of predictions **GP** classifier is confident in, and accuracy when questionable predictions are discarded.

| | Base Acc. | Confident % | Confident Acc. |
|---|---|---|---|
| MNIST | 0.9905 | 0.9804 | 0.9962 |
| n-MNIST-A | 0.9351 | 0.8569 | 0.9820 |
| n-MNIST-B | 0.9592 | 0.9377 | 0.9850 |
| n-MNIST-C | 0.7222 | 0.5345 | 0.9023 |

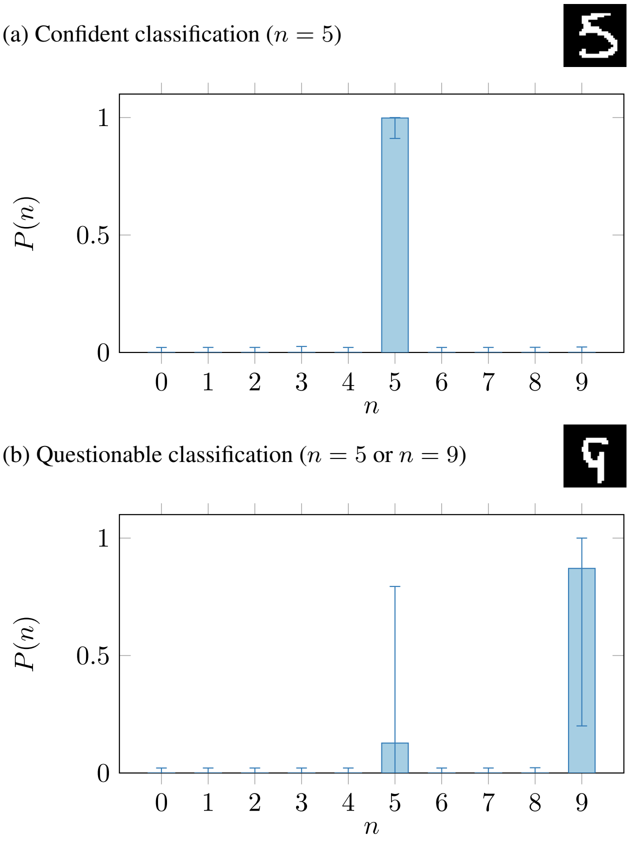
but struggles with the AWGN of n-MNIST-A, and even more so when the contrast is also reduced as in n-MNIST-C. Adding a white noise kernel to the basic polynomial kernel significantly improves the resilience of the SGPMC model at a slight cost for the noiseless datasets. Even this setup still fails to equal the overall performance of SVGP. SVGP models are therefore used for the remainder of the investigation.

### 3.3. Classification Confidence

The immediate use case for the prediction variances provided by the GP classifier is to determine whether a classification is significant. This is easily achieved for a given confidence interval, here we consider the 95% confidence interval, so take a range of $2\sigma$ around the prediction. This is visualised for two example images in Fig. 3.

Fig. 3a shows a very confident classification for an image containing a 5, matching what we would expect based on the image. Fig. 3b shows a prediction for class 9, which is sensible based on the image, though with a signifiant degree of uncertainty, such that the most likely class may in fact be 5. This is presumably because the loop of the 9 is not complete resulting in some level of ambiguity.

*Figure 3.* Predicted class probabilities and 95% confidence bounds (clipped to $[0, 1]$) for two example items from the MNIST dataset.



(a) Confident classification ($n = 5$)

(b) Questionable classification ($n = 5$ or $n = 9$)

Table 3 shows the percentage of predictions the GP classifier is confident in for each dataset, i.e., those for which the lower bound of the highest class probability does not overlap the upper bound of any other class probability. The right hand column gives the accuracy if the questionable predictions (those for which the classifier is not confident) are discarded. The accuracy in this case is significantly higher than the base accuracy for the full dataset. This indicates that in general if the classifier is confident then the prediction is more likely to be correct, and when the classifier is unsure then the prediction is more likely to be incorrect (i.e., this measure of confidence is indeed meaningful).

The variance provided by the GP is entirely dependent on

*Table 4.* Percentage of predictions **CNN** classifier is confident in, and accuracy when questionable predictions are discarded.

| | Base Acc. | Confident % | Confident Acc. |
|---|---|---|---|
| MNIST | 0.9906 | 0.9737 | 0.9977 |
| n-MNIST-A | 0.9293 | 0.6982 | 0.9956 |
| n-MNIST-B | 0.9588 | 0.7882 | 0.9962 |
| n-MNIST-C | 0.7200 | 0.1606 | 0.9950 |

the prediction probability; it is simply defined as $\sigma^2 = p - p^2$. Applying this directly to the CNN prediction probabilities is therefore considered. The results are given in Table 4 and are noticeably different to those for the GP classifier. The CNN is confident in significantly fewer predictions (just 16% for n-MNIST-C), but is almost always correct for the classifications in which it is confident (over 99.5% in all cases).

This concept therefore seems to provide some meaningful information for both classifiers, with an unsurprising trade-off between the proportion of 'confident' classifications and the accuracy of those predictions. The results for the CNN are perhaps more desirable, though usability is drastically reduced. The GP makes a significant number of incorrect, confident predictions, but retains better usability and is Bayesian in its derivation. These factors are explored further in Sec. 4.1. Increasing the range considered for significant GP classifications to a value greater than $\pm 2\sigma$ might improve GP prediction significance.

### 3.4. Variance Distribution

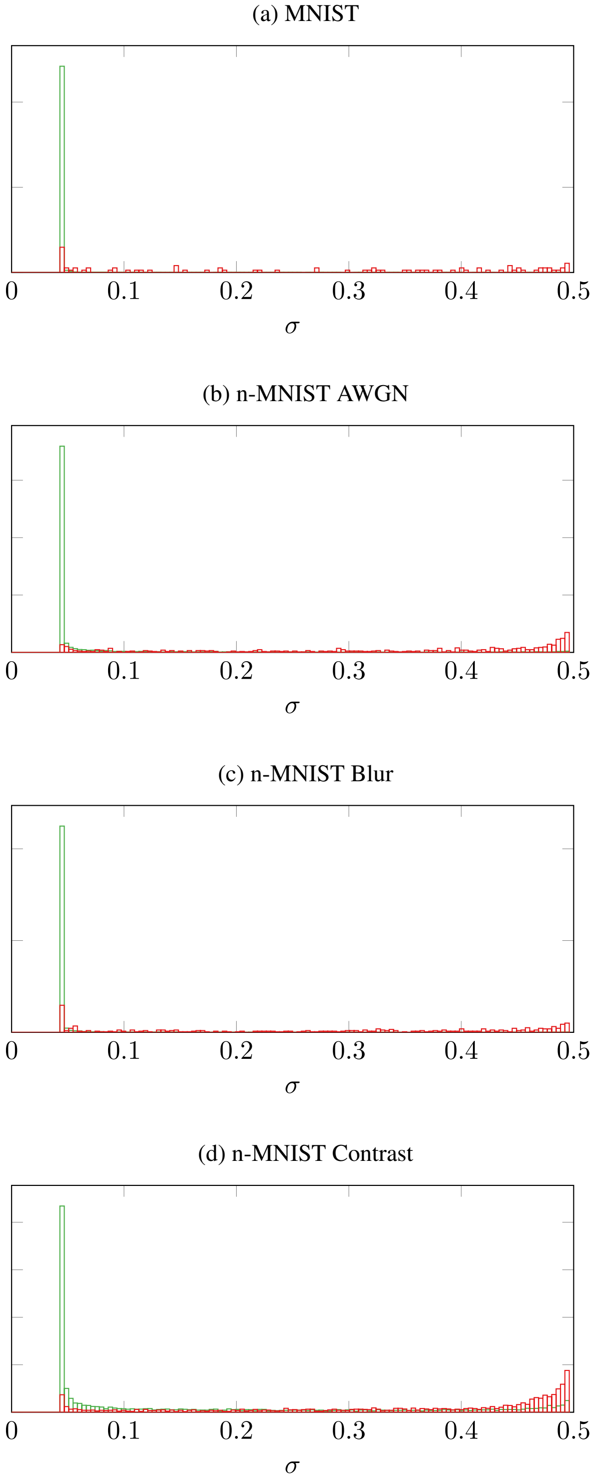
Confidence can also be inferred based on the distribution of predicted variances for correct and incorrect classifications. These distributions are shown for each dataset in Fig. 4. It is clearly visible, as suggested above, that the predicted variance does correlate to the accuracy of each prediction. Correct predictions are typically very confident (i.e., have a very low variance), while incorrect classifications typically have a larger, less predictable variance.

From these distributions it is also possible to estimate confidence intervals. For predicted class, $C_p$, actual class, $C_a$, and some standard deviation, $\gamma$:

$$P(C_p = C_a \mid \sigma < \gamma) \approx \left(1 + \frac{P(\sigma < \gamma \mid C_p \neq C_a)}{P(\sigma < \gamma \mid C_p = C_a)}\right)^{-1}$$

A confidence value for a specific prediction can, therefore, be predicted from its standard deviation. Or an estimate of $\sigma$ for some given confidence interval can be made using an iterative process. For the above datasets, the 95% confidence intervals are given by $\sigma \approx 0.04354$, $\sigma \approx 0.04724$, $\sigma \approx 0.04353$ and $\sigma \approx 0.04370$ respectively. The variability in these values indicates that, while these measures may

*Figure 4.* Distribution of predicted $\sigma$ for correct (green) and incorrect (red) classifications for MNIST and n-MNIST datasets.


(a) MNIST

(b) n-MNIST AWGN

(c) n-MNIST Blur

(d) n-MNIST Contrast

*Figure 5.* Example CleverHans perturbed images ($\epsilon = 0.15$).

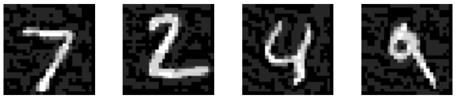


*Figure 6.* Classification performance and confidence for CleverHans perturbed images.

| | |
|---|---|
| CNN Acc. | 0.7352 |
| GP Acc. | 0.7477 |
| Confident % | 0.7246 |
| Confident Acc. | 0.8293 |

be valid for that specific test set, they are not transferable; this is discussed further in Sec. 4.1

A similar procedure can be carried out using prediction probabilities directly, and equivalently for CNN prediction probabilities. Both these techniques yield less discriminatory distributions, so any inferred confidence is significantly less useful.

### 3.5. Adversarial Perturbations

Adversarial perturbations (AP) are explored in detail in (Bradshaw et al., 2017), but there is little consideration of the of AP effect on confidence. Here AP for the MNIST test set are generated using the CleverHans (Papernot et al., 2017) implementation of the fast gradient sign method (Goodfellow et al., 2014). A value of $\epsilon = 0.15$ is chosen to generate visually insignificant alterations, as shown in Fig. 5.

These perturbations, though, have a significant impact on classification performance (given in Fig 6). The GP classifier slightly improves on CNN accuracy, though it is clear that the alterations to the input images cause significant confusion. Comparing with n-MNIST-C in Table 3, it is apparent that the number of predictions for which the classifier is confident is much higher for AP (72% compared with 53%), but the accuracy when only looking at the confident predictions is much lower (83% compared with 90%). The GP classifier is therefore selecting incorrect classes with high confidence more often when considering AP, so provides significantly less benefit in terms of confidence.

This is supported by the distributions shown in Fig. 7, which are significantly less separable than those for MNIST and n-MNIST. Taking $\sigma < 0.04353$, which provides 95% confidence for n-MNIST-C above, the confidence is now reduced to just 87%. It seems, therefore, that AP which fool the CNN also succeed in fooling the GP classifier.

*Figure 7.* Distribution of predicted $\sigma$ for CleverHans perturbed images.

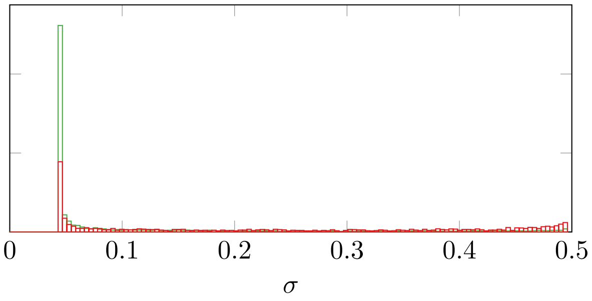


*Figure 8.* Regression performance metrics for the r-MNIST test set.

| | CNN | GPR |
|---|---|---|
| RMSE | 0.1320 | 0.1043 |
| CORR | 0.9740 | 0.9835 |

### 3.6. Regression

For the regression task described in Sec. 2.4 the GP slightly outperforms the CNN, both in terms of RMSE and correlation, as shown in Table 8. The predicted $\sigma$ values, though, provide little information as to the accuracy of a given prediction as demonstrated by the lack of any correlation in Fig. 9.

There is, therefore, little information about confidence gained for this regression task. While the predicted variance does not correlate to error in this case, it is information that is not provided at all by a CNN. It may be the case that for some tasks variance and error do correlate and so can be used to determine a meaningful confidence measure for predictions.

*Figure 9.* Absolute error against standard deviation for all 9820 images in the r-MNIST test set.

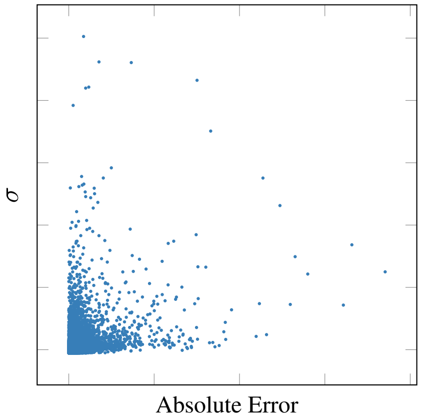
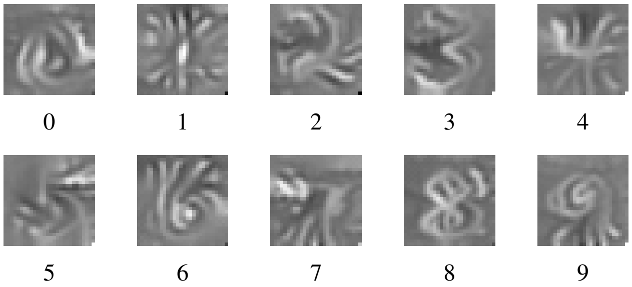


*Figure 10.* Keras-vis generated images for each number 0–9.



# 4. Discussion

## 4.1. How confident can we be?

Sec. 3 demonstrates that, for classification at least, GPs seem to provide useful information about the confidence of their predictions. We are then faced with the question of how confident we can be in these confidence measures.

First, another form of adversarial image are considered; these are inspired by the demonstration that CNNs can be easily fooled using nonsensical images (Nguyen et al., 2015). Images which maximise the CNN output activation for each class are generated using Keras-Vis (Kotikalapudi & contributors, 2017), shown in Fig. 10. While some resemble the numbers they are derived from, these images are clearly very different to the MNIST images used for training, and in a number cases unrecognisable.

Unsurprisingly, the GP classifier also predicts the associated classes with very high confidence despite the differences to the MNIST training images. Continuing this avenue of investigation, completely nonsensical images are input to the classifier. For images containing entirely ones and entirely zeros, the classifier is not confident in its prediction (the desired behaviour), but images containing random noise pose a problem.

Out of 1000 such noise images, the hybrid classifier makes confident predictions in 43% of cases. Clearly this number of confident predictions for images that contain no structure resembling numeric digits whatsoever is highly undesirable and largely undermines the purpose of the method as a whole. Using the same process for the CNN as initially described in Sec. 3.3 gives more promising results, with just 8.8% of classifications of nonsense images being confident, suggesting that the hybrid GP classifier may actually be less effective in this case.

As demonstrated in Sec. 3.4, confidence measures for one dataset cannot be transferred reliably to a dataset with even very similar characteristics. If the training and test sets encompass the full scope of input data then the confidence measures produced would likely prove valuable. For real-world systems this is usually very hard to verify, so would limit the feasibility of applying this method for any useful purpose. After all, it is perhaps better to have system which provides no confidence measure, and is therefore viewed with scepticism, than a system which can make an incorrect prediction with a high confidence.

## 4.2. Future Work

There is a lot of scope for further work on CNN confidence using GPs, and potentially the CNNs directly, or other classifiers. Evaluating the technique on problems which more closely resemble real-world tasks and the medical applications described in Sec. 1.1 is clearly an important step towards wider adoption. Most large CNNs use a greater number of nodes in the final hidden layer, and have a larger number of classes, both of which may affect the feasibility of using a GP classifier.

There is also work to be done in improving the resilience of the classifier to adversarial and nonsense inputs. This is critically important for practical applications, as any datasets used for training and testing are almost certain to exclude some edge cases which need to be identified as problematic if they occur once the model is deployed.

Regression tasks are another area requiring improvement. Prediction variance would be very useful if it could be demonstrated to give some real indication of the accuracy that can be expected for a specific example.

## 4.3. Conclusion

This paper presents an investigation into CNN classification confidence estimation using GPs. A hybrid classifier using a GP trained on features from the final hidden layer of a CNN is used to make classifications which carry an associated confidence. The classifier is evaluated on the MNIST dataset and a number of variants incorporating different distortions.

The hybrid classifier is shown to perform equally or better than the CNN on which it is based. The predicted variance values are demonstrated to be meaningful in determining whether a given prediction is significant, as well as in estimating confidence intervals based on the distribution of predicted standard deviation. A brief exploration of CNN-only classification confidence is shown to be effective for some techniques, and adversarial perturbations are found to have a significant negative impact on confidence estimations.

A similar confidence estimation approach is evaluated for a simple regression task and shown to be markedly less effective than for classification. Finally, the implications of the confidence measures obtained for the MNIST dataset are discussed in the context of real-world application, and the limitations highlighted.

## Acknowledgements

Thanks to Turner Stone Ltd. for provision of the NVIDIA GeForce GTX 1080 Ti and associated hardware used for the training and evaluation of Keras and GPFlow models.

## References

Basu, S, Karki, M, Ganguly, S, DiBiano, R, Mukhopadhyay, S, and Nemani, R. Learning sparse feature representations using probabilistic quadtrees and deep belief nets. *CoRR*, abs/1509.03413, 2015.

Bishop, C. *Neural networks for pattern recognition*. Oxford university press, 1995.

Bradshaw, J, Matthews, A G de G, and Ghahramani, Z. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

Chollet, F et al. Keras. https://github.com/keras-team/keras, 2015.

Cordts, M, Omran, M, Ramos, S, Rehfeld, T, Enzweiler, M, Benenson, R, Franke, U, Roth, S, and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE conf. on computer vision and pattern recognition*, pp. 3213–3223, 2016.

De Vleaux, R, Schumi, J, Schweinsberg, J, and Ungar, L. Prediction intervals for neural networks via nonlinear regression. *Technometrics*, 40(4):273–282, 1998.

Efron, B. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pp. 569–593. Springer, 1992.

Gal, Y. and Ghahramani, Z. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. *ArXiv e-prints*, June 2015.

Garcia-Garcia, A, Orts-Escolano, S, Oprea, S, Villena-Martinez, V, and Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, December 2014.

Hensman, J, Matthews, A G. de G., Filippone, M, and Ghahramani, Z. Mcmc for variatinoally sparse gaussian processes. In *Proc. of NIPS*, 2015a.

Hensman, J, Matthews, A G. de G., and Ghahramani, Z. Scalable variational gaussian process classification, 2015b.

Heskes, T. Practical confidence and prediction intervals. In *Advances in neural information processing systems*, pp. 176–182, 1997.

Hwang, JT and Ding, A. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92 (438):748–757, 1997.

Kamnitsas, K, Ledig, C, Newcombe, V, Simpson, J, Kane, A, Menon, D, Rueckert, D, and Glocker, B. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2018/02/08 .

Khosravi, A., Nahavandi, S., Creighton, D., and Atiya, A. F. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, 22(9):1341–1356, Sept 2011. ISSN 1045-9227.

Kingma, D and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kotikalapudi, R and contributors. keras-vis. https://github.com/raghakot/keras-vis, 2017.

Lecun, Y, Bottou, L, Bengio, Y, and Haffner, P. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86:2278 – 2324, 12 1998.

Li, B, Zhang, T, and Xia, T. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

Litjens, G et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2018/02/08 .

MacKay, D. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992.

Matthews, A G. de G. et al. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18 (40):1–6, apr 2017.

Milletari, F., Navab, N., and Ahmadi, S. A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth Int'l. Conf. on 3D Vision (3DV)*, pp. 565–571, Oct 2016.

Nair, V and Hinton, G. Rectified linear units improve restricted boltzmann machines. In *Proc. of the 27th int'l. conf. on machine learning (ICML-10)*, pp. 807–814, 2010.

Nguyen, A, Yosinski, J, and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

Nix, D and Weigend, A. Estimating the mean and variance of the target probability distribution. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE Int'l. Conf. On*, volume 1, pp. 55–60. IEEE, 1994.

Papernot, N et al. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017.

Rasmussen, C. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pp. 63–71. Springer, 2004.

Sermanet, P. and LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In *The 2011 Int'l. Joint Conf. on Neural Networks*, pp. 2809–2813, July 2011.

T, Michalis. Variational learning of inducing variables in sparse gaussian processes, 16–18 Apr 2009.

Wallach, I, Dzamba, M, and Heifets, A. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855, 2015.