Gaussian Process Classifiers & CNN Uncertainty

Jonas Tjomsland
Department of Computer Science
Cambridge University
LE49 - Probabilistic Machine Learning Project
jt732cam.ac.uk

Abstract

Convolutional Neural Networks (CNN) are achieving impressive behaviour on image classification tasks. They are outperforming humans in various problems using their capability to find patterns in high-dimensional feature space. However, CNNs lack a proper measure of confidence in their predictions and tends to extrapolate too confidently when classifying data from a different underlying distribution than the one they are trained on. Gaussian Processes (GPs), on the other hand, are well known for enabling better uncertainty estimation of predictions, with the drawback of being restricted to smaller datasets. In this project, we combine the representational power of CNNs with GPs ability to tell whenever a prediction is uncertain and apply the hybrid CNN+GP version to the an image classification task. Exploring adversarial perturbations, we confirm that the hybrid model show good calibrated uncertainties on adversarial examples.

1 Introduction

Deep Neural Networks, and specifically CNNs, are today showing beyond human level of performance in a wide variety of tasks, ranging from mole classification in health care applications [1] to Atari video games [2]. However, when such Machine Learning (ML) methods are applied to real world applications, problems regarding the robustness of their decision-making arises. For humans to trust decision-making machines a measure of confidence must be given together with their recommendations. On top of this, deep learning models are vulnerable to adversarial perturbations [3], increasing the need to quantify the certainty in the predictions they make. In an ideal world, for unseen data, decision and recommendation systems based on ML models should not only supply a prediction, but also indicate "how" unfamiliar the given data is. Hence, stating its confidence in the output. Traditional neural network models does not do this [4].

Substantial work has already been conducted to extend the representation of model uncertainty in deep learning, Yarin Gal covers multiple Bayesian methods for this in his PhD thesis [5]. It has also been shown that infinitely wide deep networks can be derived as the exact equivalence as a Gaussian Process and therefore be used to leverage the uncertainty estimates that follows with that [6].

In this project, we follow the approach of Bradshaw et al. [7] and try to create a hybrid CNN + GP model before applying it to various adversarial challenges using the MNIST dataset [8]. We explore and evaluate how the predictive mean and the standard deviation given by the GP can assist in quantifying the uncertainty of a model.

2 Background

2.1 Deep Learning

State of the art deep learning methods used for image classification are today leveraging convolutional layers to capture spatial information [9, 10]. These models are delivering impressive results when it comes to accuracy, but are also vulnerable as they do not provide uncertainty information. A deep learning model is trained to learn point estimates of its parameters. For classification problems, this often leads to a prediction represented by a probability vector which also is a point estimate. Gal mentions this [5], stating that using the *Softmax* function on such point estimates can result in overconfident predictions on data far from the training data's underlying distribution. This inherent weakness of traditional deep models is also what allows adversarial attacks to "fool" models into giving high confident miss-classifications and hence the reason that the probability vector resulting from the *Softmax* cannot be interpreted as "true" confidence of the model.

2.2 Gaussian Processes

Gaussian Processes [11] enables an approach to measure the predictive uncertainty. A GP is a generalisation of the better known Gaussian probability distribution where a distribution over functions are constructed instead of a distribution over random variables. It is, as opposed to a neural net, a non-parametric approach and like other Bayesian methods it starts out with a prior distribution, and updates its beliefs based on observed data, leading to a posterior distribution over functions. GPs are fully defined by a mean $m(x)$ and a covariance function $k(x, x')$, often called kernel. This represent the mean of the distribution at point x and the covariance between x and x' respectively. Using this analogy, different kernels are used to explain different types of data where the key is that if x and x' are determined by the kernel to be closely related then the output of the functions at these point will be similar. It is the ability to learn distributions over predictions instead of point estimates that allows GPs to easier estimate the uncertainty in outputs. Later, we explore how the variance of the predictions can be used as such a measure.

2.2.1 Gaussian Process Classification

For the binary classification case, where the number of classes is 2, $\mathbf{C} = 2$, the underlying idea is to place a GP prior over the latent functions $f(\mathbf{x})$ and then "squash" this through the logistic function. By doing this we compute a prior as follows,

$$\pi(\mathbf{x}) = p(y = +1|\mathbf{x}) = \sigma(f(\mathbf{x})). \tag{1}$$

Inference can then be implemented by first obtaining the distribution of the latent variable with respect to a test point,

$$p(f_*|X, \mathbf{y}, \mathbf{x}_*) = \int p(f_*|X, \mathbf{x}_*, \mathbf{f})p(\mathbf{f}|X, \mathbf{y})d\mathbf{f}, \tag{2}$$

followed by computing the probabilistic prediction as,

$$\bar{\pi}_* = p(y_* = +1|X, \mathbf{y}, \mathbf{x}_*) = \int \sigma(f_*)p(f_*|X, \mathbf{y}, \mathbf{x}_*)df_*. \tag{3}$$

The integral in equation (2) can unfortunately not be computed analytically due to the non-Gaussian likelihood. Similarly, some sigmoid functions can lead to equation (3) being analytically intractable, although in the binary case this can often be solved by simple numerical techniques.

In the multi-class classification approach we construct $\mathbf{C}$ latent functions with independent Gaussian priors, one for every class. There are multiple methods for multi-class approximation that scales to big data sets, among them, Laplace and Expectation Propagation [12, 13] as well as variational inference. In this project, *stochastic variational inference* [14] is used, enabling the GP to handle the large amount of data given to it. The method uses inducing points to approximate the true GP posterior with a GP conditioned on a subset of the data. Scalability has normally been seen as one of GPs weaknesses, a problem due to the $\mathcal{O}(N^3)$ complexity that arises from the inversion of the covariance matrix.

3 Method

The implemented hybrid model in this projects consist of two main parts. First, a traditional CNN trained on the MNIST dataset, followed by a Gaussian Process model. The outputs from the top level feature layer of the CNN, before the *softmax* classifier, is extracted and given as input data to the GP. This way the CNNs ability to capture information in high dimensional feature space is used to output a lower dimensional representation of the images, which then can be analysed by the GP to allow for uncertainty estimation.

3.1 CNN architecture and training

The CNN used to create the low dimensional representations of the image data has three convolutional layers, followed by a pooling layer, dropout and finally two fully connected layers before the *softmax* function. It is trained on the complete MNIST training set, 60k samples, using 3 epochs. The obtained test accuracy was 0.9877. Following this, a copy of the model was made where the last *softmax* layer removed. The resulting modified model had a 128 dimensional output.

3.2 GP training

For the GP, the *stochastic variational inference* model from GPflow is used [15]. This enables sparse GPs using the inducing points method implemented by Hensman et al. [14]. All training samples are ran through the trained, modified CNN described above. Of the resulting 60k, 128 dimensional samples, 1000 are used as training points for the GP. We use 200 inducing points. The Matern32 kernel, with added independently and identically normally-distributed noise, is implemented and we use the multi-class likelihood function supplied by GPflow. Using the Adam optimizer [16], the GP is trained for 5000 epochs resulting in a evidence lower bound loss (ELBO) of -1160.96.

3.3 Adversarial examples and perturbations

To explore and evaluate the ability of the hybrid model to estimate uncertainty in predictions we investigate the performance of the model on 5 different types of modified data:

A) MNIST with added additive white gaussian noise

B) MNIST with added motion blur

C) MNIST with a combination of added additive white gaussian noise and reduced contrast

D) MNIST after an adversarial attack intended to "fool" the model

E) MNIST rotated by different angles

In the following result and discussion sections this different data types are referred to as A-E. Below, image examples of data type A-D are presented.

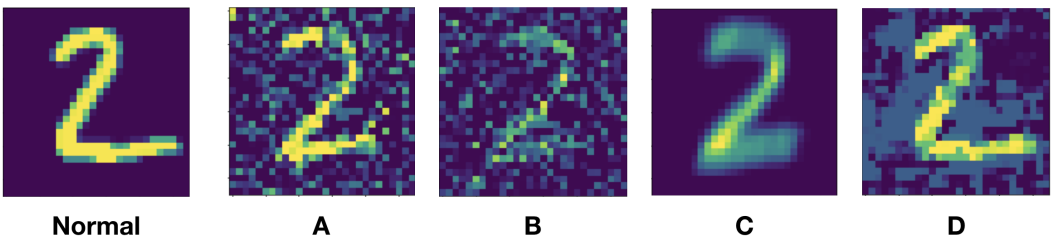

Figure 1: The different versions of the MNIST data sets used in this work

4 Results & Discussion

4.1 Accuracy

Although we in this project are mostly interested in investigating the variance and thus the uncertainty given by the hybrid model, we also compare the accuracy obtained on the different types of data. In the table below, all accuracies except case E are presented. We also compare the accuracy obtained with this work's hybrid version (CNN+GP) to that of the unmodified original CNN.

Table 1: Accuracies for the CNN and this work's hybrid CNN+GP model for different testing data

	Original MNIST	A	B	C	D
Accuracy CNN	0.9899	0.9323	0.7703	0.9426	0.0892
Accuracy CNN+GP	0.9846	0.9464	0.7391	0.8933	0.1135

As expected, we observe a lower accuracy on the adversarial examples than on the original MNIST test set. The difference in accuracy, although small for some cases, can be explained by the different underlying distribution that generates the modified images compared to that of the data the model is trained on. We also notice that the adversarial attack example are able to "fool" the model to such a degree that it classifies almost everything wrong. However, note that to the human eye, images from data type D are still quite straightforward to classify.

4.2 Uncertainty estimation

In this section we investigate the variance of the prediction distribution provided by the CNN+GP model and whether it can be used as a measure of uncertainty. In a perfect scenario, when giving the model data generated from an underlying distribution different from the one we trained it on, it should be able to indicate this uncertainty aspect, ultimately telling us that these inputs are far from something it has seen before. The mean variance of the distributions over latent functions might assist us with this, ideally it should be low for familiar data and higher for unfamiliar data. We look at the image examples where the same image is rotated before classification. In Figure 2 the standard deviation of the prediction distribution with respect to the orientation of the input image is presented.

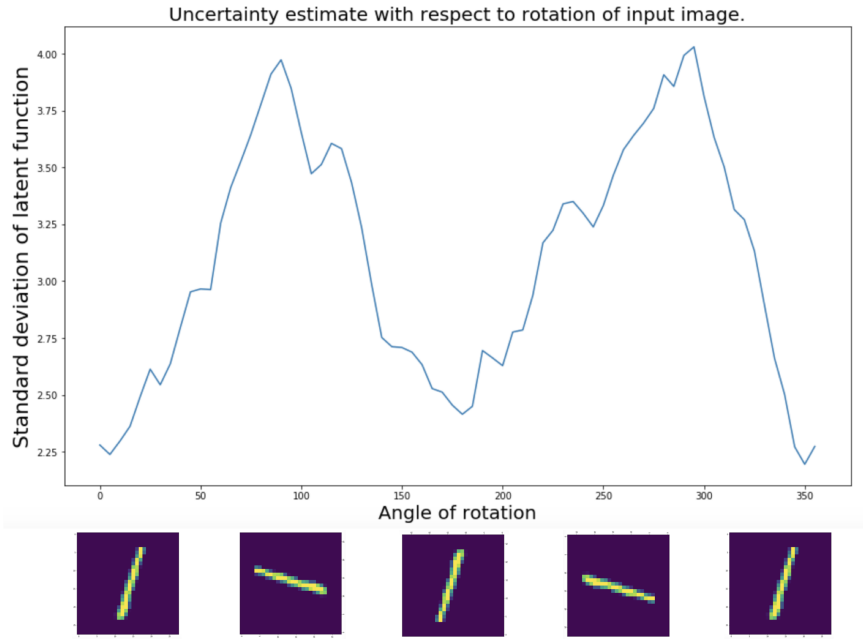

Figure 2: The standard deviation of the prediction distribution with respect to image rotation.

We observe that the standard deviation in Figure 2 increases as we start rotating the image, hence move away from images generated by the underlying distribution of the training data. The neat example with an image of the number one leads to the 180 degrees rotation being familiar and thus results in a low standard deviation again.

To further investigate the effect unfamiliar data has on variance we look at the adversarial examples. In Table 2, the variance of the predicted class, averaged over all testing samples, is presented for the different types of test data.

Table 2: Mean variances for the predicted class for different types of test data.

	Original MNIST	A	B	C	D
Variance	0.0929	0.1873	0.2076	0.1650	0.2062

Interestingly, the variances from Table 2 confirms the above mentioned hypothesis. For the original MNIST test set the variance is very low on average, indicating that this is images similar to what it knows and that it can be certain in its decision. For the other data sets the results seems to show that the variance gives an indication of the unfamiliarity of the test data. We also observe that the variances presented in Table 2 seems to correlate with the accuracies in Table 1, which is logical. One would expect similarity between test data and training data to be proportional with model performance.

Further, we investigate the difference in variance between correct and wrong predictions for the different data sets. If the variance is a good measure for uncertainty, we would expect the model to be more certain on average for cases which it predicts correctly than for cases it miss-classifies. In Table 3 such an comparison is presented.

Table 3: Mean variance for predicted classes for correct and wrong guesses respectively.

	Original MNIST	A	B	C	D
Mean variance for correct guesses	0.0960	0.1969	0.2060	0.1628	0.1465
Mean variance for wrong guesses	0.2258	0.2118	0.1685	0.2036	0.1465

The results from the comparison in Table 3 varies. For the data types where high accuracy was achieved, Original, A & C, there is a clear tendency that the variance is lower when the model is correct than when it's wrong. This indicates that in these cases, the variance relates to the uncertainty of the decision. For data type, B, one of the cases where the model performs worse, the variance is actually lower for the wrong predictions than for the correct one. For the adversarial attack example there are no difference between the two variances, this is probably due to the fact that the model is completely wrong in it's guessing

4.3 Reject option

We have shown results indicating that the model is able to report lower confidence for images outside its experience using the variance of the prediction distribution. The question now is, how should this be incorporated in a classifier such that it has the ability to quantify its confidence, maybe even output "I don't know" if it is too uncertain. A commonly used approach is the concept of a reject option. It is based on the idea that miss-classifications often would occur when the maximum class probability, $max_j p(\mathcal{C}_j|\mathbf{x})$, is relatively low. Knowing this, we can require that $max_j p(\mathcal{C}_j|\mathbf{x})$ is above some threshold Θ for the decision to be allowed. Alternatively, more applicable to multi-class classification, we can implement the same threshold idea, but now require that the difference between the most probable and second most probable class must be above it. Nevertheless, neither of these include the variance or the indication it provides about operating with unfamiliar data. However, the same threshold approach could be applied. Take a driver-less car operated by a deep reinforcement learning (DRL) agent. If the DRL method incorporates the uncertainty estimation capabilities of GPs investigated in this project it should be able to set a threshold on the variance of the prediction distributions. In that way, when the variance is above the threshold, a human operator could be notified that the agent is experiencing very unfamiliar data and needs assistance in the decision making.

5 Limitations & Future work

This work has its limitations. Firstly, the choice of covariance function and its impact on the models ability to report confidence is not investigated thoroughly. For GPs, the design and complexity of the kernels play a major role in how well the data can be represented. It would be interesting to further investigate this aspect, doing a quantitative analyses of different kernels effect one the uncertainty estimates.

Secondly, we do not dive deep into how the variance of the prediction distribution given by the presented CNN+GP model can be used to actively handle adversarial attacks. Even though Bradshaw et al. cover aspects of this [7], they only consider that GPs have better calibrated uncertainties than CNNs and how their predicted probabilities are more robust. An approach to countermeasure adversarial attacks, specifically looking at the variance as an uncertainty estimator, seems to be unexplored.

6 Conclusion

In this work we have successfully combined a deep learning method's ability to create representations of high dimensional data with the uncertainty estimation benefits that Gaussian Processes enables. We show that the hybrid model performs on similar level as the baseline CNN with respect to accuracy. While in the meantime giving better calibrated confidence in its prediction and a way of telling us when it is given data that is too far from what it has seen before. The results show that a hybrid version of a CNN and a GP does work in practice and indicates that for real world applications, where a measure of confidence is crucial, this method could be applied.

References

[1] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

[5] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[6] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[7] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

[8] Yann LeCun, Corinna Cortes, and Christopher Burges. The mnist database of handwritten digits. 1998.

[9] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[11] Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. 2006. *The MIT Press, Cambridge, MA, USA*, 38:715–719, 2006.

[12] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

[13] Carlos Villacampa-Calvo and Daniel Hernández-Lobato. Scalable multi-class gaussian process classification using expectation propagation. *arXiv preprint arXiv:1706.07258*, 2017.

[14] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[15] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo Le'on-Villagr'a, Zoubin Ghahramani, and James Hensman.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.