Kernel Density Network for Quantifying Regression Uncertainty in Face Alignment

Lisha Chen
Rensselaer Polytechnic Institute
chenl21@rpi.edu

Qiang Ji
Rensselaer Polytechnic Institute
jiq@rpi.edu

Abstract

For deep neural networks, it is important to quantify the uncertainty in its predictions. So a probabilistic neural network with a Gaussian assumption was widely used. However, in real data especially image data, the Gaussian assumption typically cannot hold. We are interested in modeling a more general distribution, e.g. multi-modal or asymmetric distribution. Therefore, a kernel density neural network is proposed. We adopt state-of-the-art neural network architecture and propose a new loss function based on maximizing the conditional log likelihood. And we show its application in face alignment. The proposed loss function achieves comparable or better performance than state-of-the-art end-to-end trainable deep learning based methods in terms of both the predicted labels and uncertainty in predictions. Moreover, it can be generally extended to many other regression problems such as Action Unit intensity estimation and face age estimation.

1 Introduction

It is very often that we have discrete labels in regression problems, e.g. face alignment, human body pose estimation, facial Action Unit intensity estimation and facial age estimation. For such problems, people either use regression framework and then discretize the continuous prediction or directly treat each discrete label as one class and use classification framework as adopted in facial age estimation [9]. The benefit of a regression framework is that it takes the relative value of the labels into consideration and naturally makes the labels with close values more difficult to distinguish from each other. While the benefit of classification framework is that it aims to find the class with the highest response, which can often be achieved without fully connected layers in convolutional neural networks thus reducing the number of parameters in the model. And a softmax cross entropy loss puts the problem into a probabilistic framework thus is able to provide uncertainty estimation.

The goal of our work is to combine the advantage of fully convolutional network [6] in specific regression problems like face alignment and the advantage of a probabilistic neural network which not only gives label prediction but also quantifies the uncertainty of the prediction. Our proposed method only change the loss function of current heatmap regression based deep learning method for face alignment. Therefore it is compatible with state-of-the-art heatmap regression based deep learning architecture for face alignment.

2 Related work

Probabilistic neural networks can be used to provide predictive uncertainty. To formulate data uncertainty, also known as aleatoric uncertainty, authors in [8, 5] proposed to model the label as a random variable or random vector and parameterize it with the neural network output. For regression tasks, it is usually assumed that the output follows Gaussian distribution [8, 5, 3]. However, in some real-world problems, the target distribution may be asymmetric or multi-modal, which the Gaussian

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

distribution cannot model. Therefore, we propose a Kernel Density Network (KDN) and apply it to face alignment.

Face alignment is to localize facial key points in a face image. Most recent deep learning based methods for face alignment [2, 12] follows the architecture of Stacked Hourglass, first proposed by [7] to solve human body pose estimation. The loss function is typically an L-2 distance between the predicted heatmap and the ground truth heatmap, specified by putting a Gaussian distribution with fixed variance around the ground truth labels [11]. This loss function can be interpreted as a deterministic softlabel regression or as minimizing the L-2 distance between a Gaussian distribution with fixed variance and the predicted distribution. Therefore the network cannot provide good uncertainty estimation for its predictions.

3 The Proposed Method

We assume target $\mathbf{y}$ is a random vector that follows $p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta})$, where $\mathbf{x}$ is the input and $\mathbf{\Theta}$ is the neural network parameter. In this way we put the problem in a probabilistic neural network framework such that $p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta})$ is parameterized by the neural network output.

3.1 Kernel Density Network

Inspired by Kernel Density Estimation [10], we use a probabilistic neural network with an infinite mixture of Gaussian distribution assumption for regression problems [4]. Under this assumption, the target distribution is expressed as

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta}) = \int p(\mathbf{y} \mid \boldsymbol{\mu}; \mathbf{\Sigma}) p(\boldsymbol{\mu} \mid \mathbf{x}; \mathbf{\Theta}) d\boldsymbol{\mu} \tag{1}$$

where $p(\mathbf{y} \mid \boldsymbol{\mu}; \mathbf{\Sigma}) = N(\boldsymbol{\mu}, \mathbf{\Sigma})$ is a Gaussian distribution with fixed covariance matrix $\mathbf{\Sigma}$ and mean $\boldsymbol{\mu}$ in the same space as $\mathbf{y}$. For landmark detection, $\boldsymbol{\mu} \in [1, m] \times [1, n]$, where m, n are the height and width of the image.

However, this integration is difficult to compute analytically. In Kernel Density Esitmation, we sample $\boldsymbol{\mu}_i$ from $p(\boldsymbol{\mu} \mid \mathbf{x}; \mathbf{\Theta})$ and use summation to approximate the integration by $p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta}) = \sum_{i=1}^{N} p(\mathbf{y} \mid \boldsymbol{\mu}_i), \boldsymbol{\mu}_i \sim p(\boldsymbol{\mu} \mid \mathbf{x}; \mathbf{\Theta})$. Here we approximate the integration with a discrete summation as shown in Equation (2), which is different from existing work of Variational Autoencoder [4].

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta}) = \sum_{i=1}^{m} \sum_{j=1}^{n} p(\mathbf{y} \mid \boldsymbol{\mu}_{ij}; \mathbf{\Sigma}) p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \mathbf{\Theta}) \tag{2}$$

where $\boldsymbol{\mu}_{ij} = [i, j]^T$ and $\sum_{i=1}^{m} \sum_{j=1}^{n} p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \mathbf{\Theta}) = 1$.

Therefore Equation (2) can be further written as

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{1}{\sqrt{(2\pi)^2 |\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{ij})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{ij})\right) p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \mathbf{\Theta}) \tag{3}$$

It is worth noting that the form of $p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Theta})$ depends on our choice of the kernel density function. If we choose a uniform kernel with range 1, it is equivalent to a likelihood for a categorical distribution where each category is the discrete regression label.

And $p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \mathbf{\Theta})$ is the output heatmap of the neural network. In this way, we only change the loss function of the neural network for face alignment problem without modifying the network structure. The goal is to maximize the conditional likelihood without assuming any specific distribution of the target, unlike widely practiced loss function which puts a fixed Gaussian heatmap around the ground truth label as the ground truth heatmap and minimize the L-2 distance between the ground truth heatmap and the predicted one. The loss function is defined as the negative log conditional likelihood. Given training data $\mathcal{D} = \{\mathbf{x}_k, \mathbf{y}_k \mid k = 1, 2, \ldots, N\}$, we minimize the loss function to get $\mathbf{\Theta}^*$ as

shown in Eq. (4).

$$\begin{aligned}\boldsymbol{\Theta}^* &= \arg\min_{\boldsymbol{\Theta}} -\sum_{k=1}^{N} \log p(\mathbf{y}_k \mid \mathbf{x}_k; \boldsymbol{\Theta}) \\ &= \arg\min_{\boldsymbol{\Theta}} -\sum_{k=1}^{N}\sum_{i=1}^{m}\sum_{j=1}^{n} p(\mathbf{y} \mid \boldsymbol{\mu}_{ij}; \boldsymbol{\Sigma}) p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \boldsymbol{\Theta})\end{aligned} \tag{4}$$

3.2 Predictive uncertainty

The proposed target distribution in Eq.(1) is composed of a mixture of Gaussian distributions, thus its covariance matrix can be computed as

$$\mathrm{Cov}[\mathbf{y} \mid \mathbf{x}; \boldsymbol{\Theta}] = \boldsymbol{\Sigma} + \sum_{i=1}^{m}\sum_{j=1}^{n} (\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_{ij})(\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}_{ij})^T p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \boldsymbol{\Theta}) \tag{5}$$

where $\bar{\boldsymbol{\mu}} = \sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{\mu}_{ij} p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \boldsymbol{\Theta})$.

The uncertainty of the prediction is quantified by the square root of the determinant of the covariance matrix $|\mathrm{Cov}[\mathbf{y} \mid \mathbf{x}; \boldsymbol{\Theta}]|^{\frac{1}{2}}$.

4 Experiment and Discussion

For training, we follow the same procedure as in [2] to make a fair comparison with the performance with the same structure and training data but different loss function.

4.1 Metrics

Normalized Mean Error (NME) Same as in [2], the NME is defined as the average point-to-point Euclidean distance between the ground truth ($\mathbf{y}_{gt}$) and predicted ($\mathbf{y}_{pred}$) landmark locations normalized by the ground truth facial bounding box size $d = \sqrt{w_{bbox} * h_{bbox}}$. n denotes the n-th testing sample.

$$\mathrm{NME} = \frac{1}{N}\sum_{n=1}^{N} \frac{||\mathbf{y}_{pred}^n - \mathbf{y}_{gt}^n||_2}{d^n} \tag{6}$$

Area under the Curve (AUC) Based on the NME in the test dataset, we can draw a Cumulative Error Distribution Curve with NME as the horizontal axis and percentage of test images as the vertical axis. Then the AUC is computed as the area under that curve for each test dataset.

4.2 Prediction Accuracy

We perform tests on 300W test dataset and Menpo Challenge dataset. The result of the softlabel is from [2]. While the paper did not provide the NME, we compute it from the trained model provided by the author [1]. For the Gaussian assumption, since it is not directly compatible with heatmap regression based method, we use ResNet-50 with two outputs, mean and variance. The results are shown in Table 1, where KDN-Uniform refers to the KDN using a uniform kernel, which is equivalent to softmax cross entropy. And KDN-Gaussian refers to the KDN using a Gaussian kernel.

Comparison of different loss

Compared to the probabilistic neural network that predicts mean and covariance for Gaussian distribution, we are able to predict a more flexible distribution including multi-mode and asymmetric distributions. And rather than providing a variance for uncertainty quantization, the probability map from our algorithm is able to give more information such as the boundary of the face, and the regions that are supposed to be other landmarks.

Compared to the softlabel loss which puts a 2D Gaussian with fixed covariance and use the L-2 distance for the training, the effect is that they will approximate a Gaussian distribution with different

Table 1: Prediction accuracy on 300W Test and Menpo Challenge dataset

Method	300W Test		Menpo Challenge	
	NME (%)	AUC (%)	NME (%)	AUC (%)
Gaussian	2.92	57.6	2.67	59.4
Softlabel[2]	2.56	66.9	2.32	67.5
KDN-Uniform	2.57	66.3	2.33	67.3
KDN-Gaussian	2.49	67.3	2.26	68.2

means but the same covariance, so that the heatmap predictions hardly give any information on their uncertainty, neither in the level of landmarks nor samples.

Compared to the KDN-uniform loss, which is equivalent to the cross entropy loss in classification to find the most possible pixel location. And this method eliminates the spatial correlation information given by the pixel location value but treat each class as independent. This method can only learn the spatial relationship of each class from the data.

4.3 Uncertainty Quantification

In Fig.1 , we plot the estimated uncertainty versus the prediction error for each landmark point rather than average error over all landmarks in a sample image which better illustrates uncertainty prediction for each landmark. We can see that the predicted uncertainty is highly correlated with the prediction error. While the softlabel loss is not designed for uncertainty estimation, the heatmap it predicts still to some extent reflects prediction uncertainty. Compare to the softlabel loss, our method gives a better uncertainty estimation.

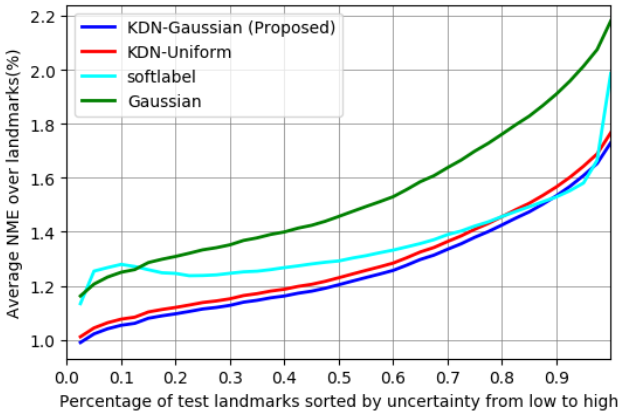

Figure 1: Predicted uncertainty v.s. prediction error on 300W testset

Visualization

Fig. 2 demonstrates that the proposed method can distinguish between occluded uncertain landmarks and non-occluded landmarks based on the predicted heatmap. For occluded landmarks, the predicted heatmap usually has a flatter shape than the non-occluded ones. While the traditional softlabel regression methods can hardly demonstrate the predictive uncertainty in occluded landmarks.

Fig. 3 demonstrates that the proposed method can capture distribution with a more flexible shape. For landmarks lie on the facial boundary, the predicted heatmap usually has a shape along the local edge of the face. While the traditional softlabel regression method still predicts a circular shape that represents a standard 2D Gaussian.

Figure 2: Sample heatmaps generated from two methods for **occluded** landmarks (better viewed in color). The 1st row is the proposed kernel density method, the 2nd row is the softlabel method. The displayed landmarks are subsets of the 68 points, the first 7 columns show point 1,5,9,13,17 and the last 3 columns show point 31,46,37,49,55.

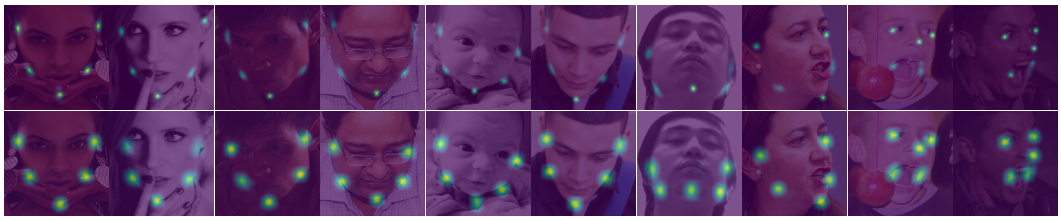

Figure 3: Sample heatmaps generated from two methods with **flexible distribution shape** (better viewed in color). The 1st row is the proposed kernel density method, the 2nd row is the softlabel method. The displayed landmarks are subsets of the 68 points, the first 8 columns show point 1,5,9,13,17 and the last 2 columns show point 31,46,37,49,55.

4.4 Sensitivity to Bandwidth

Similar as in Kernel Density Estimation, the choice of bandwidth $\mathbf{\Sigma}$ will affect the smoothness of the final predicted distribution. Larger $\mathbf{\Sigma}$ gives smoother predicted distribution.

In this work, we choose $\mathbf{\Sigma} = \sigma I$. And we analyze the sensitivity to the choice of σ in Table 2. Based on our observation, the prediction accuracy is not very sensitive to the choice of σ when σ is larger than 1, but the convergence speed is affected. Larger σ takes more time to converge. Too small σ converges faster but has a more similar behavior with KDN-Uniform that has a larger error and the predicted distribution is more centered. For face alignment we choose a small value $\sigma = 1$. But for other tasks such as human body pose estimation, the optimal choice for σ may be different. It is possible to model σ dependent on the input $\mathbf{x}$ and we leave it for future research.

Table 2: Sensitivity to bandwidth on 300W Test dataset

σ	NME (%)	AUC (%)
0.1	2.56	66.3
0.3	2.54	66.7
1	2.49	67.3
3	2.49	67.2
10	2.50	66.9

5 Conclusion

In this work, aiming at quantifying uncertainty for neural network predictions with asymmetric or multi-modal distribution, we propose a kernel density network inspired from kernel density estimation. By changing the loss function to maximize the conditional log likelihood, it achieves comparable or slightly improved performance on the testing dataset. Moreover, the predicted probability map is able to quantify predition uncertainty as well as capture more general distributions than Gaussian.

Besides that, the KDN with Gaussian kernel is able to avoid or reduce the downsampling error comparing to the softlabel loss or the KDN with a uniform kernel. Because often the heatmap is 4 times smaller compared to both the width and height of the original input image. This will lead to the downsampling error which makes it difficult to distinguish between the locations of two very close but different landmarks. Our method constructs a continuous mixture of 2D Gaussian distribution from the predicted heatmap. Therefore during testing, we are able to find the mode of the continuous distribution even if it lies between two pixels.

Future work will focus on making use of the probability map in other tasks such as occlusion detection and boundary detection. It would also be interesting to apply this method to other regression tasks such as facial age estimation and facial Action Unit intensity estimation.

References

[1] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). `https://adrianbulat.com/face-alignment`, accessed 02-November-2018.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[3] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? (Nips), 2017.

[4] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, page arXiv:1312.6114, December 2013.

[5] Quoc Le, Alex Smola, and Stéphane Canu. Heteroscedastic Gaussian Process Regression. *Proceedings of the 22nd international conference on Machine learning ICML 05*, 227:489–496, 2005.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440, 2015.

[7] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 483–499, 2016.

[8] D.A. Nix and A.S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pages 55–60 vol.1, 1994.

[9] Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-Variance Loss for Deep Age Estimation from a Face. *CVPR 2018*, pages 5285–5294.

[10] Jeffrey S. Simonoff. *Smoothing Methods in Statistics*. Springer, 1996.

[11] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 1799–1807, Cambridge, MA, USA, 2014. MIT Press.

[12] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018.

Appendix

A Implementation Details

Since the difference between the softlabel, softmax and KDE method is only the loss function, to make a fair comparison, we set the random seed to be the same for each method and train each of them until convergence. We follow the implementation of [2]. We set the initial learning rate to 10^{-4} and kept it for 15 epochs and used a minibatch of 10. Then we dropped the learning rate to 10^{-5} and to 10^{-6} after another 15, training for a total of 40 epochs. We also applied random augmentation: flipping, rotation (from -50° to 50°), color jittering, scale noise (from 0.8 to 1.2) and random occlusion.

B Equations

B.1 Gradient of the proposed loss function

To demonstrate the benefits of the proposed loss function, we compute the gradient of the loss w.r.t. the layer before softmax. To simplify the notation, let $w_{kij} = p(\mathbf{y}_k \mid \boldsymbol{\mu}_{ij}; \boldsymbol{\Sigma})$. Denote the layer before softmax for the sample k as f_{kij}, and the layer after softmax as p_{kij}, $p_{kij} = softmax(f_{kij})$. The derivative of the loss contributed by a training sample $\{\mathbf{x}_k, \mathbf{y}_k\}$ can be computed as

$$\frac{\partial Loss_k}{\partial f_{kij}} = \frac{p_{kij}\big(w_{kij} - \sum_{a=1}^{m}\sum_{b=1}^{n} w_{kab}p_{kab}\big)}{\sum_{a=1}^{m}\sum_{b=1}^{n} w_{kab}p_{kab}} \tag{7}$$

where $w_{kij} > 0$ measures the similarity between each pixel location in the heatmap and the ground truth landmark location. The closer the pixel location $[i, j]^T$ is to the ground truth $\mathbf{y}_k$, the higher w_{kij}. $\sum_{a=1}^{m}\sum_{b=1}^{n} w_{kab}p_{kab}$ is the expectation of the similarity over discrete probability distribution $p(\boldsymbol{\mu}_{ij} \mid \mathbf{x}; \boldsymbol{\Theta})$.

During training, if $w_{kij} > \sum_{a=1}^{m}\sum_{b=1}^{n} w_{kab}p_{kab}$, i.e. the similarity at location $[i, j]^T$ is larger than the average similarity, f_{kij} will increase. Therefore in the beginning, all the pixel locations near the ground truth (similarity greater than the average similarity threshold) will have their probability p_{ij} increased, and pixels far away (similarity smaller than the average similarity threshold) will have their probability decreased. Then the average similarity $\sum_{a=1}^{m}\sum_{b=1}^{n} w_{kab}p_{kab}$ will also increase. With the increasing average similarity, fewer pixels will have their associated probability increased. Then the heatmap will become more concentrated near the ground truth as the training process goes on.

Compared to the softmax cross entropy loss for classification, this loss takes into account the spatial location of each pixel, unlike the softmax loss that treats all the negative classes equally when performing the gradient update. More importantly, in the beginning of the training process, pixels near the ground truth will have their associated probability increased which allows for exploration around the ground truth and prevents overfitting to the ground truth.