Lightweight Probabilistic Deep Networks for Cell Segmentation

TECHNISCHE UNIVERSITÄT DARMSTADT

Probabilistic Graphical Models: Bonus Project

Christoph Reich
https://christophreich1996.github.io
christoph.reich@bcs.tu-darmstadt.de

Motivation

Introduction

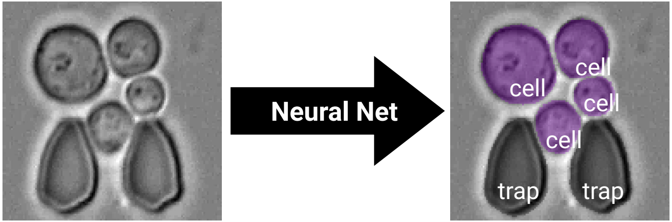

Figure: Trapped yeast cell segmentation example, adapted from [Prangemeier et al., 2020a].

- Goal: Segment cells and traps with uncertainty

Motivation

Introduction

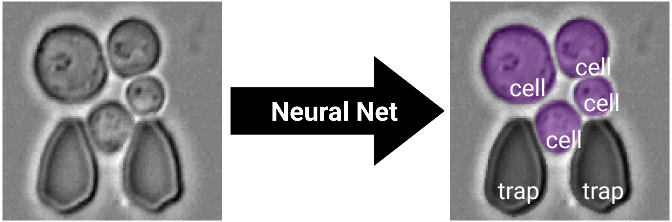

Figure: Trapped yeast cell segmentation example, adapted from [Prangemeier et al., 2020a].

- Goal: Segment cells and traps with uncertainty
- For the sake of simplicity: semantic segmentation & aleatoric uncertainty

Bayesian Neural Networks Overview
Background

- Bayes by Backprop [Blundell et al., 2015]
- Probabilistic Backpropagation [Hernández-Lobato and Adams, 2015]
- Monte Carlo Dropout [Gal and Ghahramani, 2016]
- Deep Ensembles [Fort et al., 2019]
- Stochastic Weight Averaging Gaussian [Maddox et al., 2019]

Bayesian Neural Networks Overview

Background

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Bayes by Backprop [Blundell et al., 2015]
- Probabilistic Backpropagation [Hernández-Lobato and Adams, 2015]
- Monte Carlo Dropout [Gal and Ghahramani, 2016]
- Deep Ensembles [Fort et al., 2019]
- Stochastic Weight Averaging Gaussian [Maddox et al., 2019]
- Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

- Deep neural network definition (chain of nonlinear layers)

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}; \theta) = \boldsymbol{f}^{(l)}\left(\boldsymbol{f}^{(l-1)}\left(\dots \boldsymbol{f}^{(1)}\left(\boldsymbol{x}, \theta^{(1)}\right)\right)\right)$$

- Each activation is a deterministic point estimate

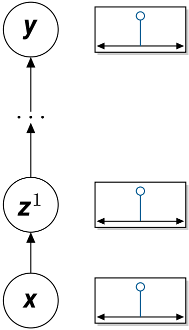

Figure: Deterministic deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]
Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Assume input to be coruppted by white Gaussian noise

$$p\left(\boldsymbol{z}^{(0)}\right) = \prod_j \mathcal{N}\left(z_j^{(0)} \,|\, x_j, \sigma_n^2\right)$$

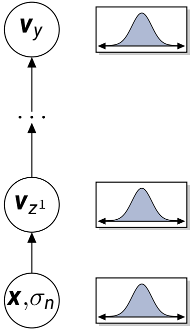

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Assume input to be coruppted by white Gaussian noise

$$p\left(\boldsymbol{z}^{(0)}\right) = \prod_j \mathcal{N}\left(z_j^{(0)} \,|\, x_j, \sigma_n^2\right)$$

  - Use ADF to fine a tractable approximation of the network activations

$$p(\boldsymbol{z}^{(0:l)}) \approx q(\boldsymbol{z}^{(0:l)}) = q(\boldsymbol{z}^{(0)}) \prod_{i=1}^{l} q(\boldsymbol{z}^{(i)})$$

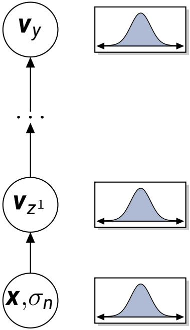

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Assume input to be coruppted by white Gaussian noise

$$p\left(\mathbf{z}^{(0)}\right) = \prod_j \mathcal{N}\left(z_j^{(0)} \,|\, x_j, \sigma_n^2\right)$$

  - Use ADF to fine a tractable approximation of the network activations

$$p(\mathbf{z}^{(0:l)}) \approx q(\mathbf{z}^{(0:l)}) = q(\mathbf{z}^{(0)}) \prod_{i=1}^{l} q(\mathbf{z}^{(i)})$$

  - Approximate subsequent activations by isotropic Gaussian

$$q(\mathbf{z}^{(i)}) = \prod_j \mathcal{N}\left(z_j^{(i)} \,|\, \mu_j^{(i)}, \sigma_j^{(i)}\right), \; \mathbf{v}_{z^{(i)}} = \left(\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}\right)$$

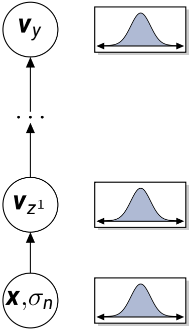

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]
Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Transform activation distribution of subsequent layers

$$\tilde{p}(\boldsymbol{z}^{(0:i)}) = p(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{z}^{(i-1)}) \prod_{j=0}^{i-1} q(\boldsymbol{z}^{(j)})$$

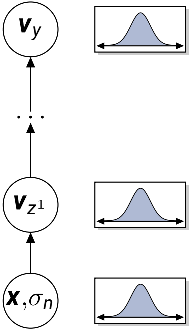

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Transform activation distribution of subsequent layers

$$\tilde{p}(\boldsymbol{z}^{(0:i)}) = p(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{z}^{(i-1)}) \prod_{j=0}^{i-1} q(\boldsymbol{z}^{(j)})$$

  - ADF performs incremental updates of variational approximation by solving

$$\underset{\tilde{q}(\boldsymbol{z}^{(0:i)})}{\operatorname{argmin}} \operatorname{KL}\left(\tilde{p}(\boldsymbol{z}^{(0:i)}) \,\|\, \tilde{q}(\boldsymbol{z}^{(0:i)})\right)$$

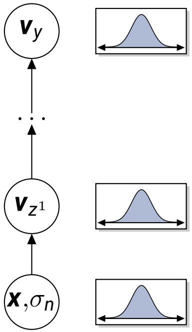

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]
Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Transform activation distribution of subsequent layers

$$\tilde{p}(\boldsymbol{z}^{(0:i)}) = p(\boldsymbol{z}^{(i)} \,|\, \boldsymbol{z}^{(i-1)}) \prod_{j=0}^{i-1} q(\boldsymbol{z}^{(j)})$$

  - ADF performs incremental updates of variational approximation by solving

$$\underset{\tilde{q}(\boldsymbol{z}^{(0:i)})}{\operatorname{argmin}} \mathrm{KL}\left(\tilde{p}(\boldsymbol{z}^{(0:i)}) \,\|\, \tilde{q}(\boldsymbol{z}^{(0:i)})\right)$$

  - Solve var. approx. by moment matching between $\tilde{p}(\boldsymbol{z}^{(0:i)})$ and $\tilde{q}(\boldsymbol{z}^{(0:i)})$.

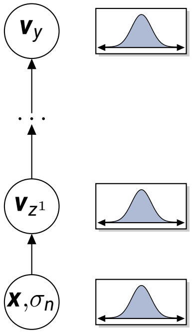

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

TECHNISCHE UNIVERSITÄT DARMSTADT

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Convert a network layer $\boldsymbol{z}^{(i)} = \boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)$ into an uncertainty propagation layer by simply matching first and second-order central moments

$$\boldsymbol{\mu}_z^{(i)} = \mathbb{E}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$

$$\boldsymbol{\sigma}_z^{(i)} = \mathbb{V}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$

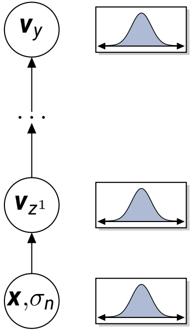

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Convert a network layer $\boldsymbol{z}^{(i)} = \boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)$ into an uncertainty propagation layer by simply matching first and second-order central moments

$$\boldsymbol{\mu}_z^{(i)} = \mathbb{E}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$
$$\boldsymbol{\sigma}_z^{(i)} = \mathbb{V}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$

  - Closed form solution available for the most common layers: Linear, Convolution, Pooling, Upsampling (Trans. Conv.), Leaky ReLU [Gast and Roth, 2018].

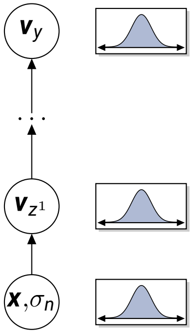

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]
Method

TECHNISCHE UNIVERSITÄT DARMSTADT

- Deep uncertainty propagation using assumed density filtering (ADF)
  - Convert a network layer $\boldsymbol{z}^{(i)} = \boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)$ into an uncertainty propagation layer by simply matching first and second-order central moments

$$\boldsymbol{\mu}_z^{(i)} = \mathbb{E}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$
$$\boldsymbol{\sigma}_z^{(i)} = \mathbb{V}_{q(\boldsymbol{z}^{(i-1)})}\left[\boldsymbol{f}^{(i)}\left(\boldsymbol{z}^{(i-1)}; \boldsymbol{\theta}\right)\right]$$

  - Closed form solution available for the most common layers: Linear, Convolution, Pooling, Upsampling (Trans. Conv.), Leaky ReLU [Gast and Roth, 2018].
- Further reading: original paper [Gast and Roth, 2018], [Murphy, 2012] & [Murphy, 2022]

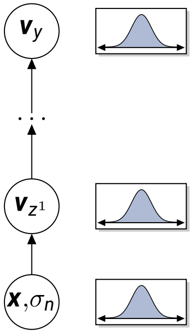

Figure: ADF-based deep neural network.

- The output of the ADF-based Lightweight Probabilistic Deep Neural Network in the semantic segmentation case is a parameterized Dirichlet distribution for each pixel

$$p(\cdot \,|\, \boldsymbol{z}) = \mathrm{Dir}(\cdot \,|\, \alpha(\boldsymbol{\mu}_z, \boldsymbol{v}_z)),\; \alpha(\boldsymbol{\mu}_z, \boldsymbol{v}_z) = \frac{\boldsymbol{m}}{s}$$

$$\boldsymbol{m} = \mathrm{softmax}(\boldsymbol{\mu}_z),\; s = c_1 + c_2 \sqrt{\sum_j m_j v_j}$$

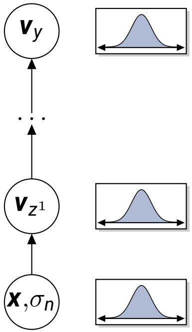

Figure: ADF-based deep neural network.

Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]

Method

- The output of the ADF-based Lightweight Probabilistic Deep Neural Network in the semantic segmentation case is a parameterized Dirichlet distribution for each pixel

$$p(\cdot \mid \boldsymbol{z}) = \mathrm{Dir}(\cdot \mid \alpha(\boldsymbol{\mu}_z, \boldsymbol{v}_z)), \ \alpha(\boldsymbol{\mu}_z, \boldsymbol{v}_z) = \frac{\boldsymbol{m}}{s}$$

$$\boldsymbol{m} = \mathrm{softmax}(\boldsymbol{\mu}_z), \ s = c_1 + c_2 \sqrt{\sum_j m_j v_j}$$

- Learning is performed by minimizing the conditional negative log-likelihood

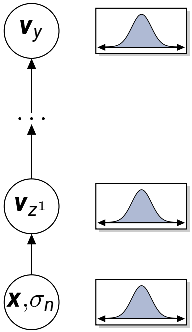

Figure: ADF-based deep neural network.

Table: Numerical results on the trapped yeast cell dataset [Prangemeier et al., 2020b].

Model	Approach	Dice ↑	IoU ↑
U-Net w/ BN [Ronneberger et al., 2015, Prangemeier et al., 2020b]	Deterministic	0.9626	0.8839
U-Net [Ronneberger et al., 2015]	ADF-based	0.9544	0.9033
DeepLabV3+ [Chen et al., 2018]	ADF-based	0.9492	0.8941

- Surprisingly, U-Net (ADF-based) slightly outperforms DeepLabV3+ (ADF-based).

Experiments

Qualitative Results ADF-Based U-Net

TECHNISCHE UNIVERSITÄT DARMSTADT

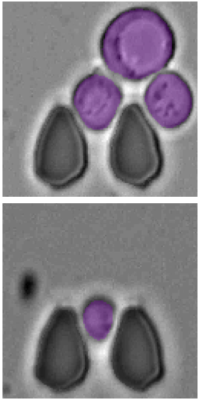

Figure: Segmentation prediction overlay.

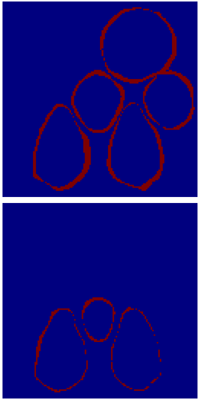

Figure: Misclassified pixels.

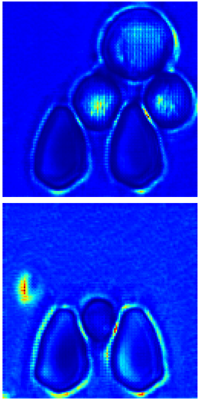

Figure: Mean uncertainty over all classes.

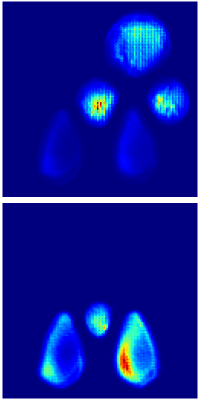

Figure: Background class uncertainty.

Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Applied Lightweight Probabilistic Deep Neural Networks to the task of cell semantic segmentation
- Lightweight Probabilistic Deep Neural Networks offer on par segmentation accuracy to the deterministic counterpart while offering uncertainties

Code will be available here:

https://github.com/ChristophReich1996

References I

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015).
Weight Uncertainty in Neural Network.
In *ICML*, pages 1613–1622. PMLR.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018).
Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.
In *ECCV*, pages 801–818.

Fort, S., Hu, H., and Lakshminarayanan, B. (2019).
Deep Ensembles: A Loss Landscape Perspective.
preprint arXiv:1912.02757.

References II

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gal, Y. and Ghahramani, Z. (2016).
  Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.
  In *ICML*, pages 1050–1059. PMLR.
- Gast, J. and Roth, S. (2018).
  Lightweight Probabilistic Deep Networks.
  In *CVPR*, pages 3369–3378.
- Hernández-Lobato, J. M. and Adams, R. (2015).
  Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks.
  In *ICML*, pages 1861–1869. PMLR.

References III

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019).
A Simple Baseline for Bayesian Uncertainty in Deep Learning.
NeurIPS, 32.

Murphy, K. P. (2012).
Machine Learning: A Probabilistic Perspective.
MIT press.

Murphy, K. P. (2022).
Probabilistic Machine Learning: An introduction (preprint already available).
MIT Press.

References IV

TECHNISCHE UNIVERSITÄT DARMSTADT

Prangemeier, T., Reich, C., and Koeppl, H. (2020a).
Attention-Based Transformers for Instance Segmentation of Cells in Microstructures.
In *IEEE BIBM*.

Prangemeier, T., Wildner, C., O. Françani, A., Reich, C., and Koeppl, H. (2020b).
Multiclass Yeast Segmentation in Microstructured Environments with Deep Learning.
In *IEEE CIBCB*.

Ronneberger, O., Fischer, P., and Brox, T. (2015).
U-Net: Convolutional Networks for Biomedical Image Segmentation.
In *MICCAI*, pages 234–241.