Paper assignment **Thijs van der Plas, DB2019 application**

I chose to evaluate the fourth paper of the list:

`Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. NeurIPS 2017.`

Question a)

- *How do authors change the NN to make it capable to estimate uncertainty for regression tasks?*
  Instead of predicting a single outcome value $\mu(\mathbf{x})$, the two parameters $\mu(\mathbf{x}), \sigma^2_{\mathbf{x}}$ of a Gaussian distribution $\mathcal{N}(\mu(\mathbf{x}), \sigma^2_{\mathbf{x}})$ are estimated. This is done by minimizing the negative log-likelihood (equation 1, page 3).

- *What is the distribution on the outputs, as defined by the NN architecture and loss?*
  The distribution on the outputs y is the gaussian:

$$p\left(y|\mathbf{x}, \mu(\mathbf{x}), \sigma^2_{\mathbf{x}}\right) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathbf{x}}}} \exp\left(\frac{-(y-\mu(\mathbf{x}))^2}{2\sigma^2_{\mathbf{x}}}\right)$$

- *What distribution on the outputs would be induced by an ensemble of such NNs?*
  For an ensemble the model changes as stated at the end of section 2 (page 5). It is an uniformly weighted ensemble of gaussians:

$$p\left(y|\mathbf{x}, \{\mu(\mathbf{x})_m\}, \{\sigma^2_{\mathbf{x},m}\}\right) = M^{-1} \sum_m \mathcal{N}(y|\mu(\mathbf{x})_m, \sigma^2_{\mathbf{x},m})$$

  which is approximated by the single gaussian:

$$p\left(y|\mathbf{x}, \{\mu(\mathbf{x})_m\}, \{\sigma^2_{\mathbf{x},m}\}\right) = \mathcal{N}(y|\mu_*(\mathbf{x}), \sigma^2_{*\mathbf{x}})$$

  with

$$\mu_*(\mathbf{x}) = M^{-1} \sum_m \mu_m(\mathbf{x})$$

$$\sigma^2_{*\mathbf{x}} = M^{-1} \sum_m \left(\sigma^2_{m\mathbf{x}} + \mu^2_m(\mathbf{x})\right) - \mu^2_*(\mathbf{x})$$

Question b)

- *What are adversarial examples?*
  Adversarial examples are training instances that are slightly distorted from their original. It is a well-known issue that some classifiers that perform well at original train/test data, fail at these (barely noticeable) distorted samples. Hence, by including adversarial examples in the training data, an algorithm becomes more robust. In the current study, adversarial examples are generated with the *fast gradient sign method* (section 2.3, page 4).

- *What is the purpose of using them to train the ensemble?*
  Because adversarial training makes the classifier robust to small perturbations, this smooths

the predictive distribution in the vicinity of the predicted labels. By use of the fast gradient sign method, this is done efficiently by considering local perturbations that would otherwise create a large loss.

- *Can an object with an unchanged prediction be an adversarial example?*
  Yes, because this would be ideal performance of the classifier: adversarial examples $(\mathbf{x'}, y)$ have the same target y as their original training example $(\mathbf{x}, y)$.

Question c)

- *Let's imagine that somebody collected a dataset with many out-of-domain images or images with wrong labels. How can the proposed uncertainty estimation method be applied to clean the dataset from such objects?*
  This uncertainty estimation predicts both the label $\hat{y}$ and the uncertainty of its prediction $p(\hat{y})$. Out-of-domain images should (if this model converged) be assigned a high uncertainty (i.e. low $p(\hat{y})$), and can thus effectively be filtered with e.g. confidence thresholding as depicted in figure 6 (page 9).