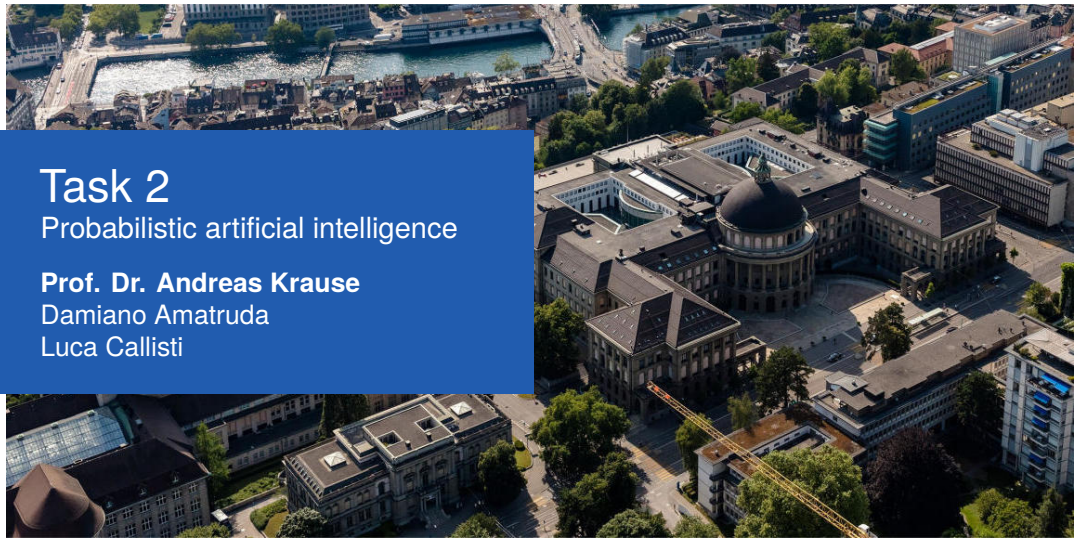
ETH zürich



Task 2

Probabilistic artificial intelligence

Prof. Dr. Andreas Krause
Damiano Amatruda
Luca Callisti

Classification problem using Bayesian Neural Networks

The dataset is composed by 60x60 RGB satellite images from various locations. Each image is classified as one or more types of land use.

- Training set: each image corresponds to exactly one out of six types of land usage.
- Test set: a certain fraction of images correspond to multiple types of land usage. Those mixed land usage types may include those not present in the training set.

Furthermore, some images might contain seasonal snow or clouds.

label 0

label 5 with cloud

label 2 with snow

label -1

Bayesian Neural Network

Bayesian Neural network learn a distribution over the weights of the network.

- Prior distribution: $p(\theta)$
- Likelihood: $p(y_{1:n}|x_{1:n},\theta) = \prod_{i=1}^{n} p(y_i|x_i,\theta)$
- Posterior distribution: $p(\theta|x_{1:n},y_{1:n}) = \frac{1}{z}p(\theta)\prod_{i=1}^{n} p(y_i|x_i,\theta)$ where $z = \int p(\theta)\prod_{i=1}^{n} p(y_i|x_i,\theta)d\theta$
- Prediction: $p(y^*|x^*,x_{1:n},y_{1:n}) = \int p(y^*|x^*,\theta)p(\theta|x_{1:n},y_{1:n})d\theta$

But usually the posterior distribution is unfeasible to calculate and it has to be approximated

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} \log\big(p(\theta)\big) + \sum_{i=1}^{n} \log\big(p(y_i|x_i,\theta)\big) \implies p(y^*|x^*,x_{1:n},y_{1:n}) = p(y^*|x^*,\hat{\theta}_{\text{MAP}})$$

MAP estimate corresponds to a point estimate of the weights, so it do not model the epistemic uncertainty.

Stochastic Weights average

1. **SWAG-Diagonal**: Starting from a pre-trained solution, run SGD and construct the posterior approximation as a independent gaussian:

$$\theta_{\text{SWAG}} = \frac{1}{T}\sum_{i=1}^{T}\theta_i, \quad \Sigma_{\text{diag}} = \text{diag}\left(\frac{1}{T}\sum_{i=1}^{T}\theta_i^2 - \theta_{\text{SWAG}}^2\right), \quad \mathcal{N}(\theta_{\text{SWAG}}, \Sigma_{\text{diag}}).$$

2. **SWAG**: Extend the idea of SWAG-Diagonal, using a low-rank plus diagonal approximation for the covariance matrix:

$$D_i = \left(\theta_i - \frac{1}{i}\sum_{j=1}^{i}\theta_j\right), \quad \hat{D} = [D_{T-k}, \ldots, D_T], \quad \mathcal{N}\left(\theta_{\text{SWAG}}, \frac{1}{2}\left(\Sigma_{\text{diag}} + \frac{1}{k-1}\hat{D}\hat{D}^T\right)\right)$$

3. **MultiSWAG:** Combine multiple independently trained SWAG approximations to produce a mixture of Gaussian approximations that approximate the posterior:

$$\theta_{(i,j)} \sim \mathcal{N}\left(\theta_{\text{SWAG}}^{(i)}, \frac{1}{2}\left(\Sigma_{\text{diag}}^{(i)} + \frac{1}{k-1}\hat{D}^{(i)}(\hat{D}^{(i)})^T\right)\right), \quad p(x) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} f(x; \theta_{(i,j)}).$$

Calibration

A model is said to be *well calibrated* if its confidence coincides with its accuracy across many predictions.
If we group the predictions into M interval bins of size $\frac{1}{M}$ according to their class probability predicted by the model, then we compare within each bin:

- how often the model thought the inputs belonged to the class:

$$\text{Confidence}(B_k) = \frac{1}{|B_k|} \sum_{i \in B_k} \hat{p}_i$$

  where $\hat{p}_i$ is the predicted probability for the predicted class of the i-th sample.

- how often the inputs actually belonged to the class:

$$\text{Frequency}(B_k) = \frac{1}{|B_k|} \sum_{i \in B_k} \mathbb{1}(\hat{y}_i = y_i)$$

  where $\hat{y}_i$ is the predicted label and y_i is the true label of the i-th sample.

If for each bin $\text{Confidence}(B_k) \approx \text{Frequencey}(B_k)$ then the model is *well calibrated*.

Results

The expected calibration error is a metric that quantifies the calibration of a model

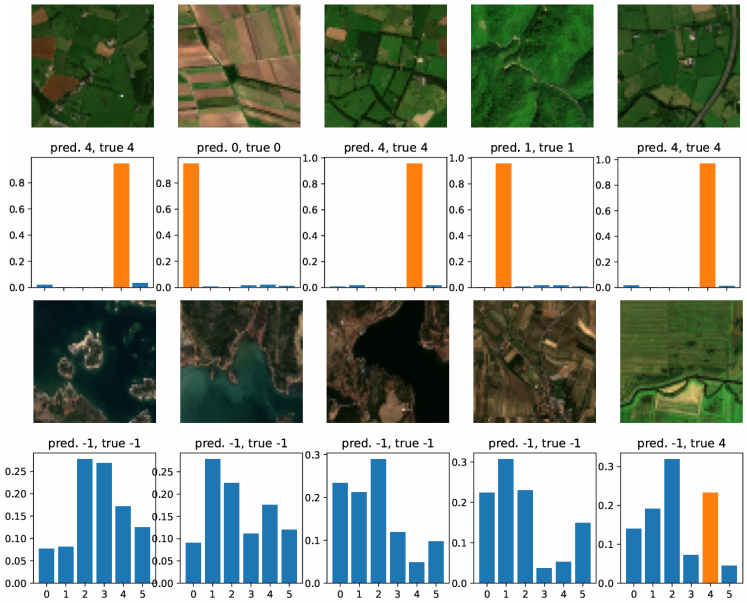
$$\text{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{N} \left|\text{Confidence}(B_k) - \text{Frequency}(B_k)\right|.$$

While for the prediction, the mean of the following function is used as cost

$$l(y, \hat{y}) = \begin{cases} 1, & \text{if } \hat{y} = -1, \\ 3, & \text{if } \hat{y} \neq -1, \text{ and } y \neq \hat{y}, \\ 0, & \text{if } \hat{y} \neq -1, \text{ and } y = \hat{y}. \end{cases}$$

The result of the model is

$$\text{cost} = 0.742, \quad \text{ECE} = 0.092.$$


pred. 4, true 4
pred. 0, true 0
pred. 4, true 4
pred. 1, true 1
pred. 4, true 4
pred. -1, true -1
pred. -1, true -1
pred. -1, true -1
pred. -1, true -1
pred. -1, true 4

Bibliography

[1] Wesley Maddox et al. *A Simple Baseline for Bayesian Uncertainty in Deep Learning*. 2019. arXiv: 1902.02476 [cs.LG]. URL: https://arxiv.org/abs/1902.02476.

[2] Andrew Gordon Wilson and Pavel Izmailov. *Bayesian Deep Learning and a Probabilistic Perspective of Generalization*. 2022. arXiv: 2002.08791 [cs.LG]. URL: https://arxiv.org/abs/2002.08791.