INITIATIVE FOR APPLIED ARTIFICIAL INTELLIGENCE

Sampling Free Epistemic Uncertainty Through Approximated Variance Propagation

appliedAI

Shaping Europe's Innovative Power in AI

A joint initiative UNTERNEHMERTUM | lpai

Aleatoric and epistemic uncertainty

- Aleatoric: Uncertainty that is intrinsic to the data generating process.
  - Joint data distribution: $\mathrm{P}(X, Y)$
  - Uncertainty about Y given X: $\mathrm{P}(Y \mid X)$
  - Cannot be reduced by observing more data.
- Epistemic: Uncertainty about data generation process.
  - E.g. the uncertainty about the parameters θ of a model of the conditional distribution
    $$p_\theta(\hat{y} \mid x)$$
  - Can be reduced by observation.
- Bayesian posterior estimate given data D:
  $$p(\hat{y} \mid x, D) = \int p_\theta(\hat{y} \mid x) p(\theta \mid D)$$

Epistemic uncertainty through noise injection

Add intermediate noise layers. Keep them active during training and inference.

During training, the model learns to compensate for the noise and hence reduces the variance in the prediction.

During inference, noise injection can be used for epistemic uncertainty estimation as the model is likely to have learned to compensate for the injected noise only within the data distribution.

Additive and multiplicative noise layer

- There are two common ways to inject noise to the input vector $X \in \mathbb{R}^n$.
- Additive: $X + Z$ for random vector $Z \in \mathbb{R}^n$
- Multiplicative: $X \circ Z$ for random vector $Z \in \mathbb{R}^n$
  - Where $\circ$ is the element wise multiplication (Hadamard product)

Dropout

- Special case of multiplicative noise.
- z is a binary noise vector where each z_i is drawn independently from a Bernoulli distribution.
- Layers with dropout: $\sigma(M(z \circ x) + b)$
- Connects noise injection with approximate Bayesian posterior inference.

Computational complexity

MC sampling scales linearly in the number of samples

Can constitute a severe bottleneck for realtime applications

However, reliable uncertainty estimates are necessary for critical applications such as autonomous cars.

Are there faster methods for estimating the uncertainty that is implicitly captured by noise injection?

Propagation of uncertainty

- How does one propagate uncertainties from already estimated quantities to derived quantities?
- Model estimate as random variable X.
- Uncertainty is modeled as covariance Σ_X.

Algebra of random variables

- R.v. $X \in \mathbb{R}^d$ *and affine map* $f: x \mapsto Mx + b$*:*

$$\Sigma_{MX+b} = M^T \Sigma_X M$$

- Independent r.v. $X, Z \in \mathbb{R}^d$ *and* $\circ$ *Hadamard product:*

$$\Sigma_{X \circ Z} = \Sigma_X \circ \Sigma_Z + E[Z]E[Z]^T \circ \Sigma_X + E[X]E[X]^T \circ \Sigma_Z$$

Non-linear transformations
Naive approach

- 1st order approximation:
  - For non-linear σ let $J_\sigma(x)$ be the Jacobian of σ.

$$\Sigma_{\sigma(X)} \approx \mathrm{J}_\sigma(\mathrm{E}[\mathrm{X}])^T \Sigma_X \, J_\sigma(E[X])$$

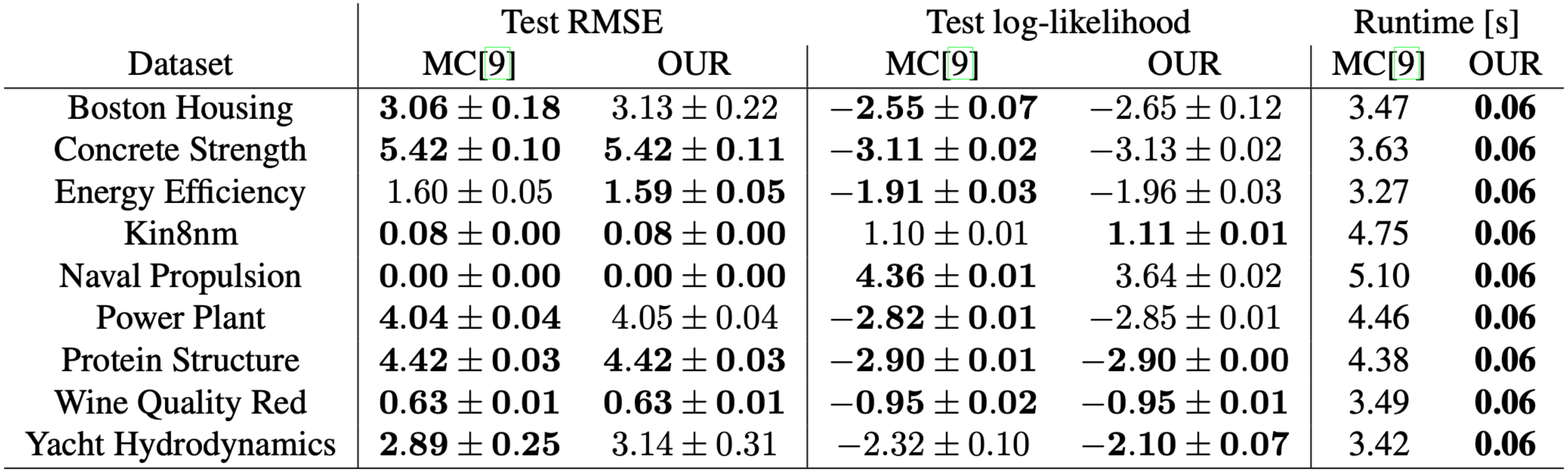
- But crude approximation

Expressive power of covariance matrices

	Covariance matrix	MC samples
Standart deviations	☑	☑
Quantiles	✗	☑
Higher-order moments	✗	☑

Method comparison

- Uncetainty propagation achievies comparable results in terms of RMSE and test likelyhood (Gaussian likelihood model).
- Runtime of MC sampling techniques scales linearly with number samples.
- Uncertainty propagation has constant runtime.



	Test RMSE		Test log-likelihood		Runtime [s]	
Dataset	MC[9]	OUR	MC[9]	OUR	MC[9]	OUR
Boston Housing	**3.06 ± 0.18**	3.13 ± 0.22	**-2.55 ± 0.07**	-2.65 ± 0.12	3.47	**0.06**
Concrete Strength	**5.42 ± 0.10**	**5.42 ± 0.11**	**-3.11 ± 0.02**	-3.13 ± 0.02	3.63	**0.06**
Energy Efficiency	1.60 ± 0.05	**1.59 ± 0.05**	**-1.91 ± 0.03**	-1.96 ± 0.03	3.27	**0.06**
Kin8nm	**0.08 ± 0.00**	**0.08 ± 0.00**	1.10 ± 0.01	**1.11 ± 0.01**	4.75	**0.06**
Naval Propulsion	**0.00 ± 0.00**	**0.00 ± 0.00**	**4.36 ± 0.01**	3.64 ± 0.02	5.10	**0.06**
Power Plant	**4.04 ± 0.04**	4.05 ± 0.04	**-2.82 ± 0.01**	-2.85 ± 0.01	4.46	**0.06**
Protein Structure	**4.42 ± 0.03**	**4.42 ± 0.03**	**-2.90 ± 0.01**	**-2.90 ± 0.00**	4.38	**0.06**
Wine Quality Red	**0.63 ± 0.01**	**0.63 ± 0.01**	**-0.95 ± 0.02**	**-0.95 ± 0.01**	3.49	**0.06**
Yacht Hydrodynamics	**2.89 ± 0.25**	3.14 ± 0.31	-2.32 ± 0.10	**-2.10 ± 0.07**	3.42	**0.06**

Comparrison of uncertainty propagation (OUR) against the original MC dropout method on various regression datasets.

Approximations in high-dimensional space

Convolutional layers

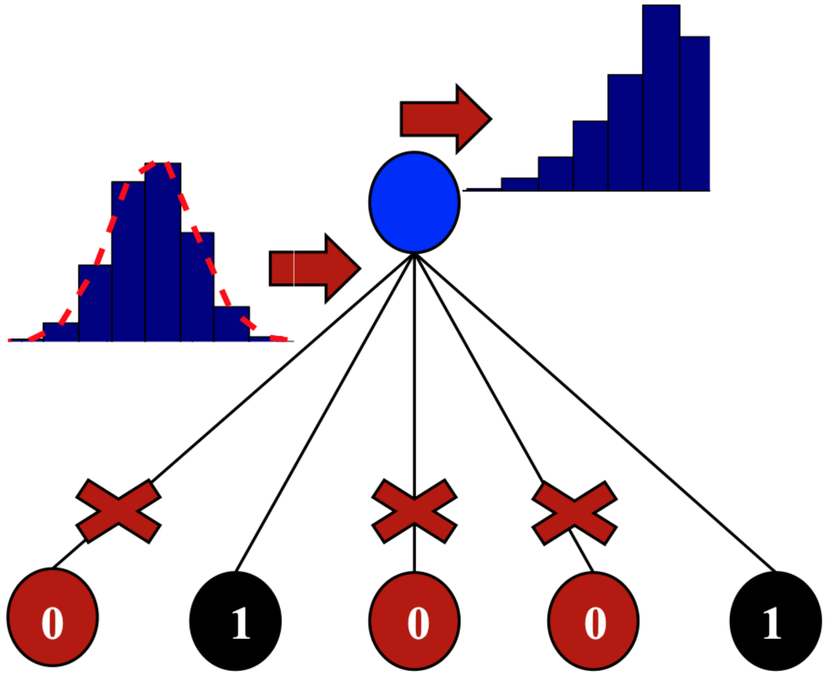
- Full covariance propagation is expensive in high dimensions, e.g. images
  - Multiplication of matrices of dimension
$$(height \cdot width \cdot channels) \times (height \cdot width \cdot channels)$$
- Convolutional layers are typically used in CV as first layers in high dimensions
- Such layers introduce relatively little covariance:
  - Kernel dimension << input dimension
  - E.g. for independent input: If two components in the output space are correlated then their corresponding kernel windows overlap.
- Only propagate variance:
  - $Var(f(X)) = main\_diagonal(\Sigma_{f(X)})$
  - Significantly reduces computational cost.

Dropout and Gaussian input approximation

- Consider the input to the activation function of a neuron trained with dropout

$$\sum_i w_i z_i x_i$$

  - x_i activations of the previous layer or input
  - w_i input weights
  - z_i independent samples of Bernoulli distribution of success probability p_i (dropout).
- Usually, the w_i are centered around 0 and x_i are unimodally distributed or come from a bounded interval.
- Under such conditions the central limit theorem predicts that the input distribution will be approximately Gaussian if the input dimension is high.



Dropout induces an approximately Gaussian distribution that is pushed through the non-linearity.

Mean and variance of neuron input distribution under dropout

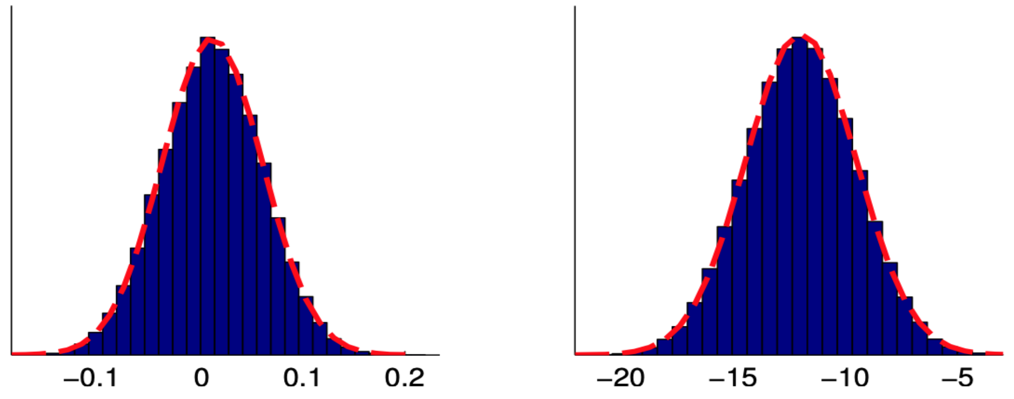
- Neuron input: $\sum_i w_i Z_i X_i$ assuming all X_i, Z_i are independent
- Mean:

$$\sum_i w_i p_i E[X_i]$$

- Variance:

$$\sum_i w_i^2 p_i (Var[X_i] + (1 - p_i) E[X_i]^2)$$

- Where $p_i = P(z_i = 1)$



Emipirical imput distribution of a hidden unit under dropout. Before (left), and after (right) training.

Variance propagation under Gaussian input distributions for ReLU

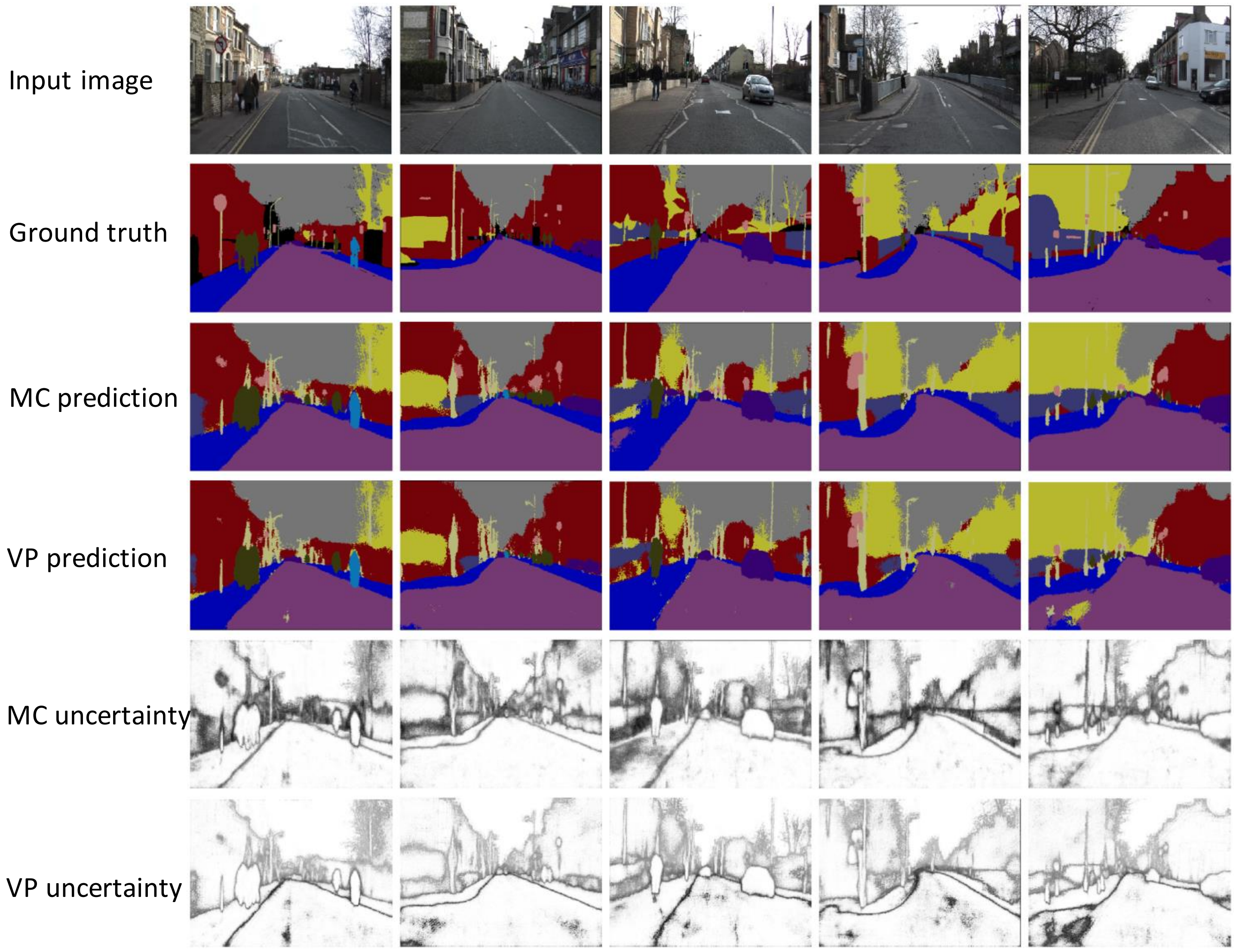
- The output variance of the ReLU function under a Gaussian input distribution can be computed analytically.
- Since only variances are propagated, one only needs to consider the one dimensional case.

$$Var_{X\sim\mathcal{N}(\mu,\sigma)}\left[\max(0,X)\right] = E_{X\sim\mathcal{N}(\mu,\sigma)}\left[\max(0,X)^2\right] - E_{X\sim\mathcal{N}(\mu,\sigma)}\left[\max(0,X)\right]^2$$

$$E_{X\sim\mathcal{N}(\mu,\sigma)}\left[\max(0,X)\right] = \int_0^\infty x\,\mathcal{N}(x;\mu,\sigma)dx$$

$$E_{X\sim\mathcal{N}(\mu,\sigma)}\left[\max(0,X)^2\right] = \int_0^\infty x^2\,\mathcal{N}(x;\mu,\sigma)dx$$

Qualitative comparison on a semantic segmentation task



Qualitative comparison on a depth regression task