Deep Neural Networks: A Bayesian Perspective

Dmitry P. Vetrov

Research professor at HSE,

Senior researcher at Yandex

Head of Bayesian methods research group

http://bayesgroup.ru

p(B|A)yesgroup.ru

Outline

- Bayesian framework in brief
- Variational inference
- Dropout as Bayesian procedure
- Sparse Variational dropout

Idea of the talk

Conditional and marginal distributions

Just to remind...

- Conditional distribution

$$\texttt{Conditional} = \frac{\texttt{Joint}}{\texttt{Marginal}}, \quad p(x|y) = \frac{p(x,y)}{p(y)}$$

- Product rule: Any joint distribution can be expressed as a product of one-dimensional conditional distributions

$$p(x,y,z) = p(x|y,z)p(y|z)p(z) = p(z|x,y)p(x|y)p(y)$$

- Sum rule: Any marginal distribution can be obtained from the joint distribution by **intergrating out** unnessesary variables

$$p(y) = \int p(x,y)dx = \int p(y|x)p(x)dx = \mathbb{E}_x p(y|x)$$

Arbitrary conditioning

- Assume we have a joint distribution over three groups of variables $p(X, Y, Z)$
- We observe Z and are interested in predicting X
- Values of Y are unknown and irrelevant for us
- How to estimate $p(X|Z)$ from $p(X, Y, Z)$?

Arbitrary conditioning

- Assume we have a joint distribution over three groups of variables $p(X, Y, Z)$
- We observe Z and are interested in predicting X
- Values of Y are unknown and irrelevant for us
- How to estimate $p(X|Z)$ from $p(X, Y, Z)$?

$$p(X|Z) = \frac{p(X, Z)}{p(Z)} = \frac{\int p(X, Y, Z)dY}{\int p(X, Y, Z)dY\,dX}$$

- Sum rule allows to build arbitrary conditional distributions at least in theory

Bayes theorem

- Conditionals inversion (follows from product rule):

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}$$

- Bayes theorem (follows from conditionals inversion and sum rule):

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx}$$

- Bayes theorem defines the rule for uncertainty conversion when new information arrives

$$\texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$

Statistical inference

- Consider standard problem of statistical inference. Given i.i.d. data $X = (x_1, \ldots, x_n)$ from distribution $p(x|\theta)$ one needs to estimate θ
- Maximum likelihood estimation (MLE):

$$\theta_{ML} = \arg\max p(X|\theta) = \arg\max \prod_{i=1}^{n} p(x_i|\theta) = \arg\max \sum_{i=1}^{n} \log p(x_i|\theta)$$

- Bayesian inference: encode uncertainty about θ in terms of a distribution $p(\theta)$ and apply Bayesian inference

$$p(\theta|X) = \frac{\prod_{i=1}^{n} p(x_i|\theta)p(\theta)}{\int \prod_{i=1}^{n} p(x_i|\theta)p(\theta)d\theta}$$

Frequentist vs. Bayesian frameworks

	Frequentist	Bayesian
Randomness	Objective indefiniteness	Subjective ignorance
Variables	Random and Deterministic	Everything is random
Inference	Maximal likelihood	Bayes theorem
Estimates	ML-estimates	Posterior or MAP-estimates
Applicability	$n \gg 1$	$\forall n$

Bayesian machine learning

- Suppose we're given training data (X, T) and a probabilistic classifier $p(t|x, W)$
- Define reasonable prior over the weights $p(W)$
- Training stage:

$$p(W|X,T) = \frac{p(T|X,W)p(W)}{\int p(T|X,W)p(W)dW}$$

- Test stage:

$$p(t^*|x^*,X,T) = \int p(t^*|x^*,W)p(W|X,T)dW$$

- Bayesian learning results in an **ensemble** of classifiers

Bayesian machine learning

- Suppose we're given training data (X, T) and a probabilistic classifier $p(t|x, W)$
- Define resonable prior over the weights $p(W)$
- Training stage:

$$p(W|X,T) = \frac{p(T|X,W)p(W)}{\int p(T|X,W)p(W)dW}$$

- Test stage:

Usually intractable

$$p(t^*|x^*,X,T) = \int p(t^*|x^*,W)p(W|X,T)dW$$

- Bayesian learning results in an **ensemble** of classifiers

Variational Bayes

- Approximate posterior with a simpler distribution from a restricted parametric family

$$p(W|X,T) \approx q(W|\phi) = \arg\min_{\phi} KL(q(W|\phi)||p(W|X,T))$$

- It can be shown that

$$\arg\min_{\phi} KL(q(W|\phi)||p(W|X,T)) = \arg\max_{\phi} \int q(W|\phi) \log \frac{p(T|X,W)p(W)}{q(W|\phi)} dW$$

- The last expression is usually denoted as $\mathcal{L}(\phi)$ and has special name **evidence lower bound** (ELBO)

Properties of ELBO

ELBO

$$\mathcal{L}(\phi) = \int q(W|\phi) \log \frac{p(T|X,W)p(W)}{q(W|\phi)} dW \to \max_{\phi}$$

has several nice properties

- We may compute its stochastic gradient by performing **mini-batching** and removing integral with its MC estimate
- We do not overfit - the richer is parametric family the closer we are to the true posterior
- We may rewrite ELBO as follows

$$\mathcal{L}(\phi) = \int q(W|\phi) \log p(T|X,W) dW - KL(q(W|\phi)||p(W))$$

Matt Hoffman, David M. Blei, Chong Wang, John Paisley. Stochastic Variational Inference. JMLR 14, 2013.

Properties of ELBO

ELBO

$$\mathcal{L}(\phi) = \int q(W|\phi) \log \frac{p(T|X,W)p(W)}{q(W|\phi)} dW \to \max_{\phi}$$

has several nice properties

- We may compute its stochastic gradient by performing **mini-batching** and removing integral with its MC estimate
- We do not overfit - the richer is parametric family the closer we are to the true posterior
- We may rewrite ELBO as follows

Regularizer

$$\mathcal{L}(\phi) = \int q(W|\phi) \log p(T|X,W) dW - KL(q(W|\phi)||p(W))$$

Data term

- The second term prevents $q(W|\phi)$ from collapsing to maximum likelihood point

New understanding of regularization

- Standard view of regularization:

  We add a regularizer to log-likelihood and try to find the weights W:

$$\log p(T|X,W) + \log p(W) \to \max_W$$

  This corresponds to MAP-estimate.

- Bayesian view on regularization:

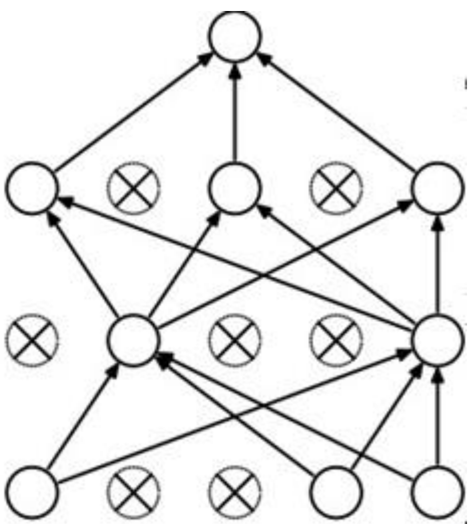
  We are searching not for the W but for the $q(W|\phi)$ thus training ensemble of networks

$$\int q(W|\phi) \log p(T|X,W) dW - KL(q(W|\phi)||p(W)) \to \max_\phi$$

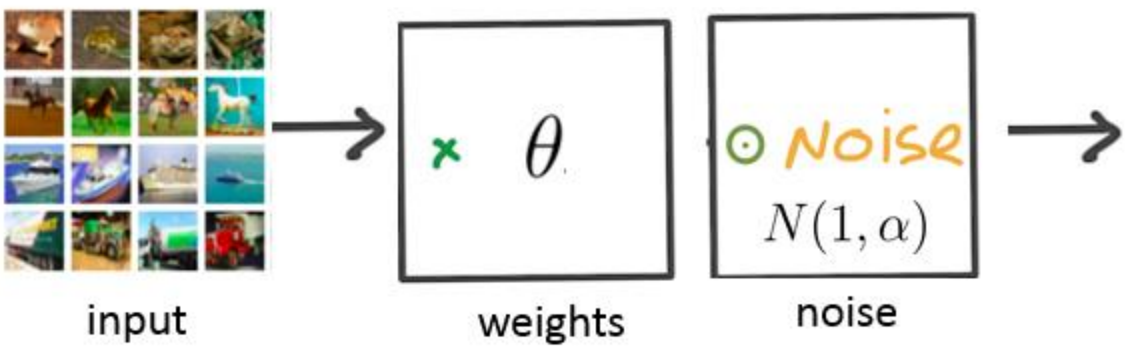
  On each iteration of training we **inject noise** in our neural network

Dropout

- Purely heuristic regularization procedure
- Inject either Bernoulli $\text{Ber}(\xi|p)$ or gaussian $\mathcal{N}(\xi|1,\alpha)$ noise to the weights during training
- The magnitude of the noise p and α respectively are set manually



Binary dropout



Gaussian dropout

Reverse engineering of dropout

- In 2015 Kingma, Salimans and Welling decided to understand the nature of dropout
- They assumed that gaussian dropout corresponds to Bayesian procedure that optimizes ELBO using SGD with $q(W|\theta, \alpha) = \mathcal{N}(W|\theta, \alpha\theta^2)$

$$\int \mathcal{N}(W|\theta, \alpha\theta^2) \log p(T|X, W) dW - KL(\mathcal{N}(W|\theta, \alpha\theta^2)||p(W)) \to \max_{\theta}$$

- The first term corresponds to the criterion that is really optimized during dropout training... BUT there is no KL-term!

Diederik P. Kingma, Tim Salimans, Max Welling. Variational Dropout and the Local Reparameterization Trick. arXiv:1506.02557

Reverse engineering of dropout

- In 2015 Kingma, Salimans and Welling decided to understand the nature of dropout
- They assumed that gaussian dropout corresponds to Baeysian procedure that optimizes ELBO using SGD with $q(W|\theta,\alpha)=\mathcal{N}(W|\theta,\alpha\theta^2)$

$$\int \mathcal{N}(W|\theta,\alpha\theta^2)\log p(T|X,W)dW - KL(\mathcal{N}(W|\theta,\alpha\theta^2)||p(W)) \to \max_{\theta}$$

- The first term correposnds to the criterion that is really optimized during dropout training... BUT there is no KL-term!
- IDEA! What if KL-term does not depend on θ?
- If one could find such prior $p(W)$ that

$$KL(\mathcal{N}(W|\theta,\alpha\theta^2)||p(W))$$

is independent from θ...

Diederik P. Kingma, Tim Salimans, Max Welling. Variational Dropout and the Local Reparameterization Trick. arXiv:1506.02557

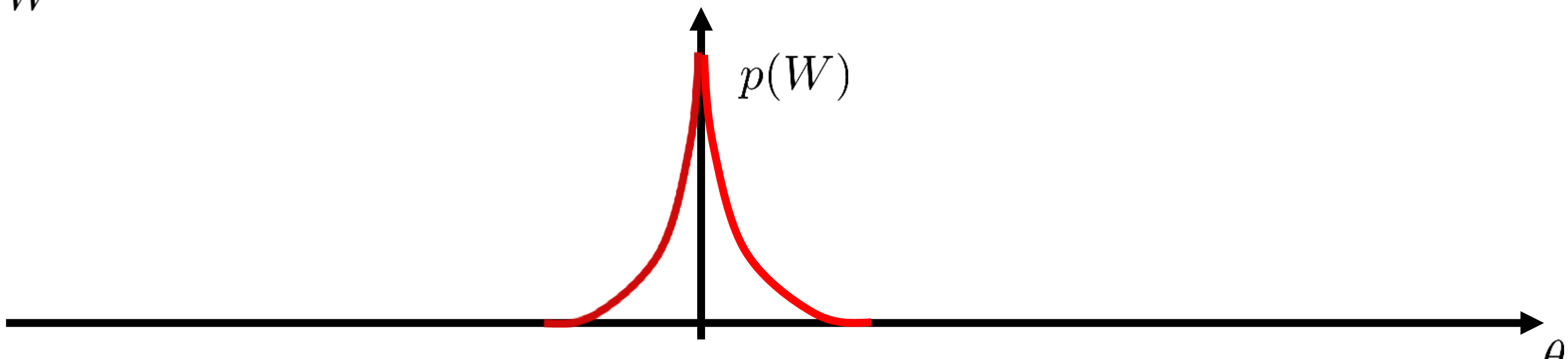
And... voila!

- They managed to find distribution $p(W)$ such that KL-term does not depend on θ
- It means that gaussian dropout really corresponds to reasonable Bayesian procedure
- Surpizingly the distribution $p(W)$ appeared to be very interpretable

$$p(W) \propto \frac{1}{|W|}$$

  known as **log-uniform** prior

- It penalizes the precision (number of significant digits) with which we find W



Variational dropout

- Remember that in gaussian and binary dropouts the magnitude of the noise is to be defined manually
- With Bayesian interpretation of dropout we have better option
- KL-term does not depend on θ but still depends on α

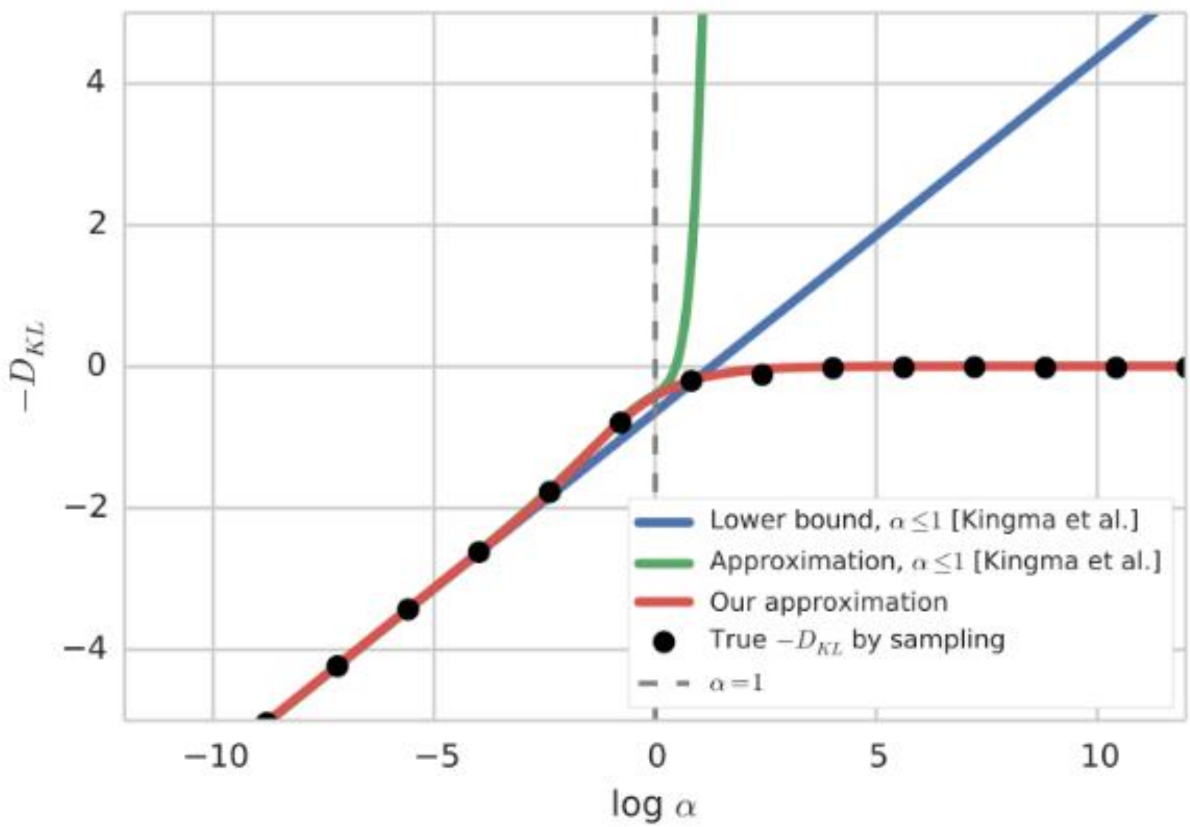
$$\mathcal{L}(\theta, \alpha) = \text{DataTerm}(\theta, \alpha) + KL(\alpha) \to \max_{\theta}$$

- Why not trying to optimize ELBO both w.r.t. θ and α?

$$\mathcal{L}(\theta, \alpha) = \text{DataTerm}(\theta, \alpha) + KL(\alpha) \to \max_{\theta, \alpha}$$

Approximation for *KL*-term

- KL-term cannot be computed in closed form
- However since this is 1-dimensional function of α_{ij} we may approximate it by analytic function



Sparse VDO

- Now we may extend the variational family even further and assign **individual dropout rates** α_{ij} per each weight

$$q(W|\theta, \alpha) = \prod_{i,j} \mathcal{N}(w_{ij}|\theta_{ij}, \alpha_{ij}\theta_{ij}^2)$$

- It can be shown that if $\alpha_{ij} \to +\infty$ then $\theta_{ij} = O\left(\frac{1}{\alpha_{ij}}\right)$ i.e.

$$\lim_{\alpha_{ij} \to +\infty} q(w_{ij}|\theta_{ij}, \alpha_{ij}) = \delta(0)$$

- Incredebly efficient way for removing the redundancy of current deep architectures
- Up to 99.9% of the weights in the layer become irrelevant

Dmitry Molchanov, Arsenii Ashukha, Dmitry Vetrov. Variational Dropout Sparsifies Deep Neural Networks. ICML 2017.

Additive noise reparameterization

- Using chain-rule we have

$$\frac{\partial \mathcal{L}}{\partial \theta_{ij}} = \frac{\partial \mathcal{L}}{\partial w_{ij}} \frac{\partial w_{ij}}{\partial \theta_{ij}}$$

- Let $q(w_{ij}|\theta_{ij}, \alpha_{ij}) = \mathcal{N}(w_{ij}|\theta_{ij}, \alpha_{ij}\theta_{ij}^2)$. Then using **reparameterization trick** yields

$$w_{ij} = \theta_{ij}(1 + \sqrt{\alpha_{ij}}\varepsilon), \quad \varepsilon \sim \mathcal{N}(\varepsilon|0, 1)$$

$$\frac{\partial w_{ij}}{\partial \theta_{ij}} = 1 + \sqrt{\alpha_{ij}}\varepsilon$$

- Huge variance when $\alpha_{ij} \gg 1$

Additive noise reparameterization

- Solution: new variable $\sigma_{ij}^2 = \alpha_{ij}\theta_{ij}^2$
- Variational distribution takes the form $q(w_{ij}|\theta_{ij}, \sigma_{ij}) = \mathcal{N}(w_{ij}|\theta_{ij}, \sigma_{ij}^2)$. Then using **reparameterization trick** yields

$$w_{ij} = \theta_{ij} + \sigma_{ij}\varepsilon, \quad \varepsilon \sim \mathcal{N}(\varepsilon|0, 1)$$

$$\frac{\partial w_{ij}}{\partial \theta_{ij}} = 1$$

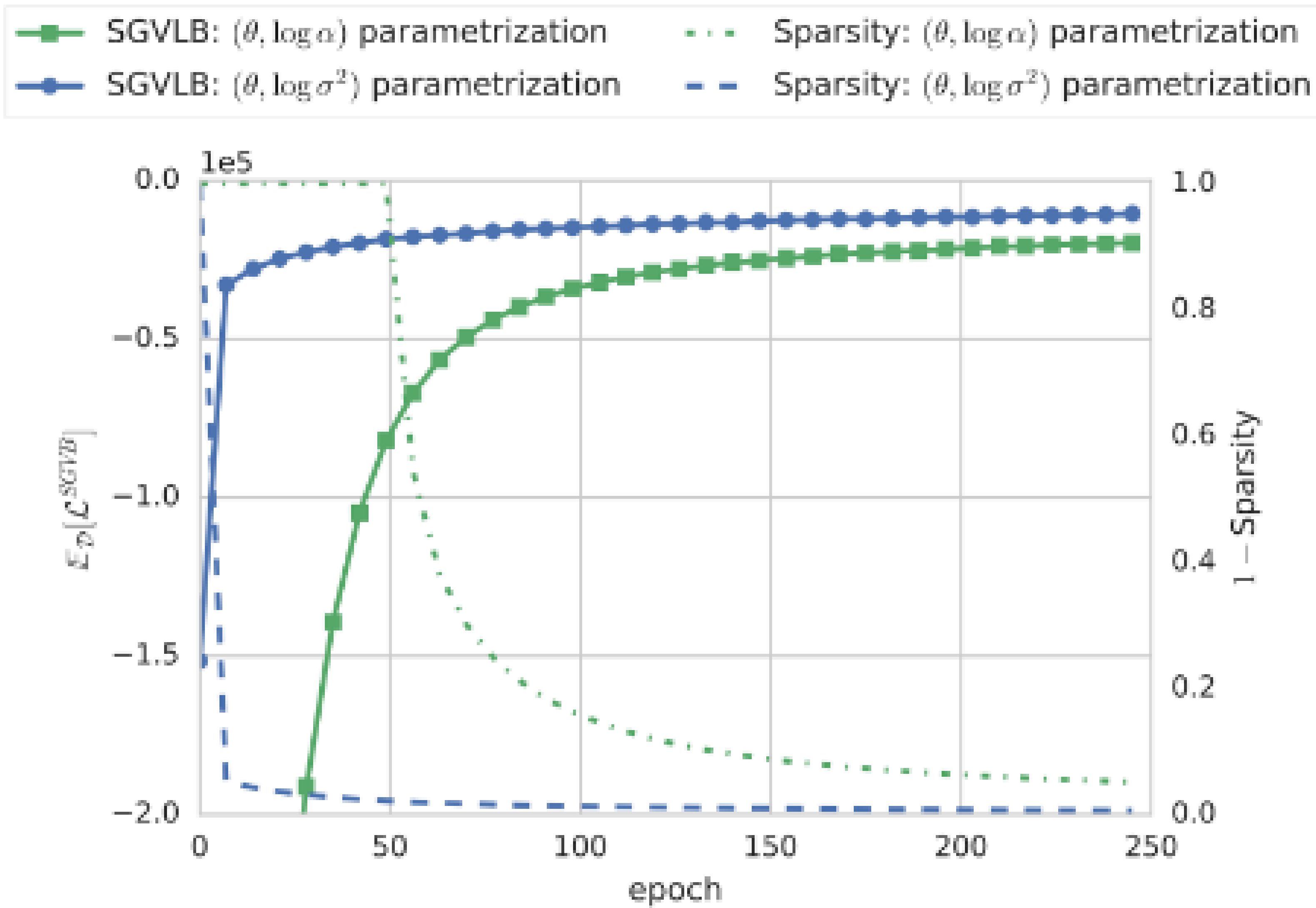
- Now KL-term becomes dependant on θ_{ij}

$$\frac{\partial KL(\alpha_{ij})}{\partial \theta_{ij}} = \frac{\partial KL(\alpha_{ij})}{\partial \alpha_{ij}} \frac{\partial \alpha_{ij}}{\partial \theta_{ij}} = -2 \frac{\partial KL(\alpha_{ij})}{\partial \alpha_{ij}} \frac{\sigma_{ij}^2}{\theta_{ij}^3}$$

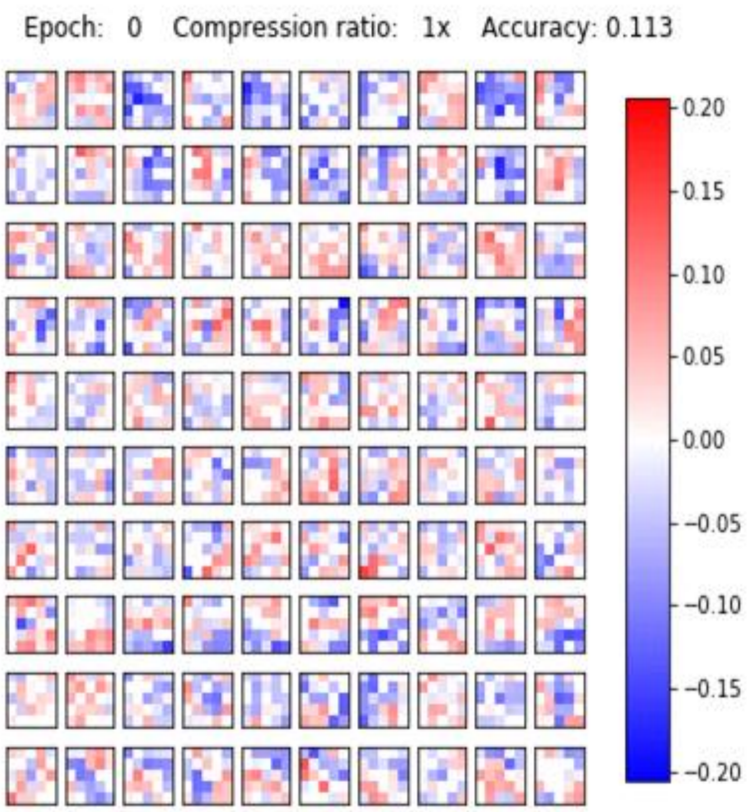
- The price to pay: we may no longer share α's between the weights

Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, Dmitry Vetrov. Structured Bayesian Pruning via Log-Normal Multiplicative Noise. arXiv:1705.07283

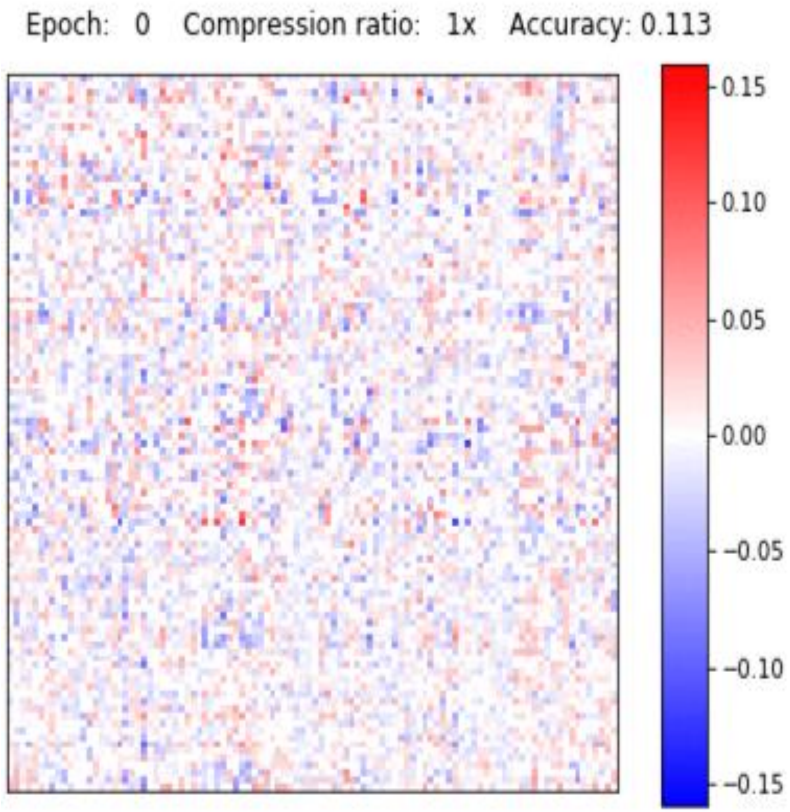
Variance reduction



Visualization



LeNet-5: convolutional layer



LeNet-5: fully-connected layer (100 x 100 patch)

Open Problems

- To develop better variance reduction methods for stochastic variational inference
- To develop stochastic optimization procedures with faster convergence rates that take into account the **structure** of optimization problem
- To find efficient ways of **Bayesian ensembling**
- To find more flexible variational families that are still memory-efficient to keep 1M-dimensional distributions

Conclusions

- Bayesian framework is extremely powerful and extends ML tools
- We do have scalable algorithms for approximate Bayesian inference
- Bayes + Deep Learning = 
- Even the first attempts of neurobayesian inference give impressive results
- Summer school on NeuroBayesian methods, August, 2018, Moscow, http://deepbayes.ru

Deep|Bayes