Learning Multiscale Stochastic Finite Element Basis Functions with Deep Neural Networks

Rohit Tripathy and Ilias Bilionis

Predictive Science Lab [*http://www.predictivesciencelab.org/*](http://www.predictivesciencelab.org/)

Purdue University

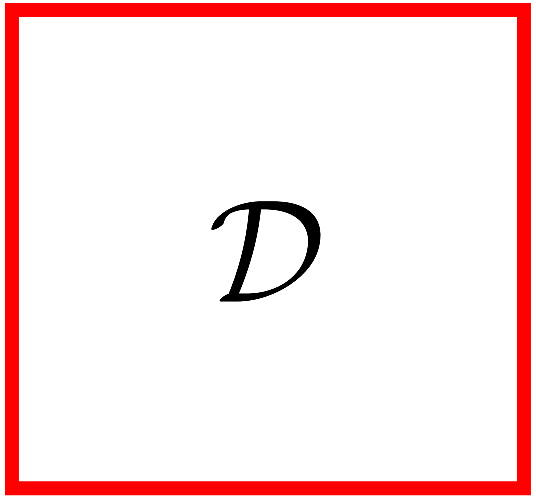
West Lafayette, IN, USA

PURDUE
UNIVERSITY

STOCHASTIC MUTLISCALE ELLIPTIC PDE

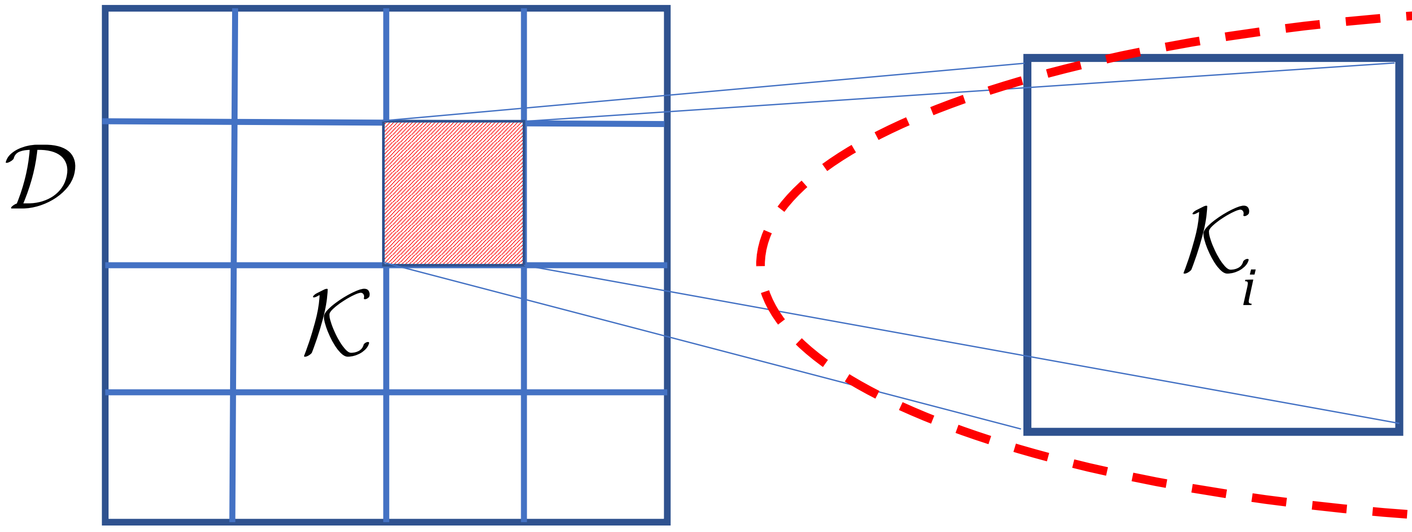
$$\boxed{-\nabla(\mathbf{a}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) = f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{D} \subset \mathbb{R}^d} \rightarrow \text{PDE}$$

$$\boxed{g(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \partial\mathcal{D}} \rightarrow \text{BC}$$



- Consider **a** to be multiscale.
- Uncertainty in diffusion field $\mathbf{a}(\mathbf{x})$,BCs $g(\mathbf{x})$,forcing function $f(\mathbf{x})$.
- We consider uncertainty only in diffusion field $\mathbf{a}(\mathbf{x})$.

MULTISCALE FEM (MsFEM)[1]



Solve:

$$-\nabla(\mathbf{a}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) = 0 \, \forall \mathbf{x} \in \mathcal{K}_i$$

$$\mathbf{u}(\mathbf{x}) = u_j \, \forall \mathbf{x} \in \partial\mathcal{K}_i$$

KEY STEP !

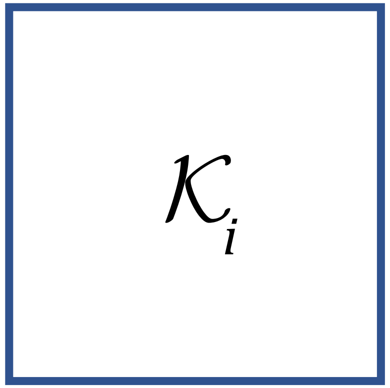
Couple adaptive basis functions with full FEM solver.

Reference:

[1]-Efendiev and Hou, *Multiscale finite element methods: theory and applications*. (2009)

PURDUE UNIVERSITY

Key Idea of Stochastic MsFEM[1]



$$
\begin{aligned}
-\nabla(\mathbf{a}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) &= 0\ \forall\mathbf{x}\in\mathcal{K}_i \\
\mathbf{u}(\mathbf{x}) &= u_j\ \forall\mathbf{x}\in\partial\mathcal{K}_i
\end{aligned}
$$

Exploit local low dimensional structure

Reference:

[1]-Hou et. al, *Exploring The Locally Low Dimensional Structure In Solving Random Elliptic Pdes*. (2016)

Curse of dimensionality

Dimension	10 points / dimension	1 second / evaluation
1	10	10 sec
2	100	~ 1.6 min
3	1,000	~ 16 min
4	10,000	~ 2.7 hours
5	100,000	~ 1.1 days
6	1,000,000	~ 1.6 weeks
…	…	…
20	1e20	3 trillion years (240x age of the universe)

CHART CREDIT: *PROF. PAUL CONSTANTINE**

* Original presentation: https://speakerdeck.com/paulcon/active-subspaces-emerging-ideas-for-dimension-reduction-in-parameter-studies-2

TECHNIQUES FOR DIMENSIONALITY REDUCTION

- Truncated Karhunen-Loeve Expansion (also known as Linear Principal Component analysis)[1].
- Active Subspaces (with gradient information[2] or without gradient information[3]).
- Kernel PCA[4]. (Non-linear model reduction).

References:

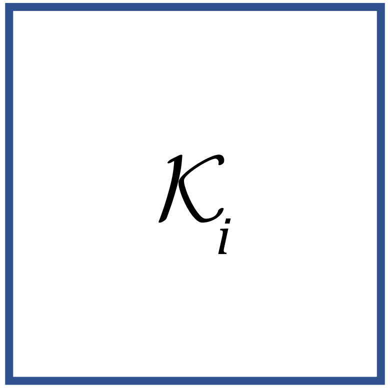
[1]- Ghanem and Spanos. *Stochastic finite elements: a spectral approach* (2003).

[2]- Constantine et. al. *Active subspace methods in theory and practice: applications to kriging surfaces*. (2014).

[3]-Tripathy et. al. *Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation*. (2016).

[4]-Ma and Zabaras. *Kernel principal component analysis for stochastic input model generation*. (2011).

This work proposes …



$$
\begin{aligned}
-\nabla(\mathbf{a}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) = 0 \; \forall \mathbf{x} \in \mathcal{K}_i \\
\mathbf{u}(\mathbf{x}) = u_j \; \forall \mathbf{x} \in \partial\mathcal{K}_i
\end{aligned}
$$

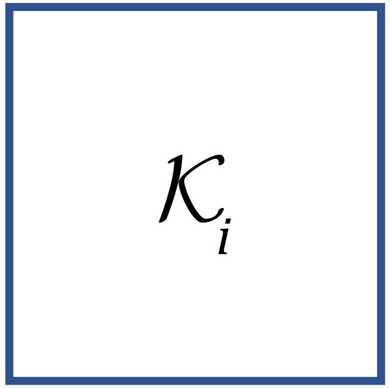
Replace the solver for this homogeneous PDE with a DNN surrogate.

WHY:

- Capture arbitrarily complex relationships.
- No imposition on the probabilistic structure of the input.
- Work directly with a discrete snapshot of the input.

- Specifically, we consider:

$$-\nabla(\mathbf{a}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})) = 0 \, \forall \mathbf{x} \in \mathcal{K}_i$$
$$\mathbf{u} = u_j \, \forall \mathbf{x} \in \partial\mathcal{K}_i$$

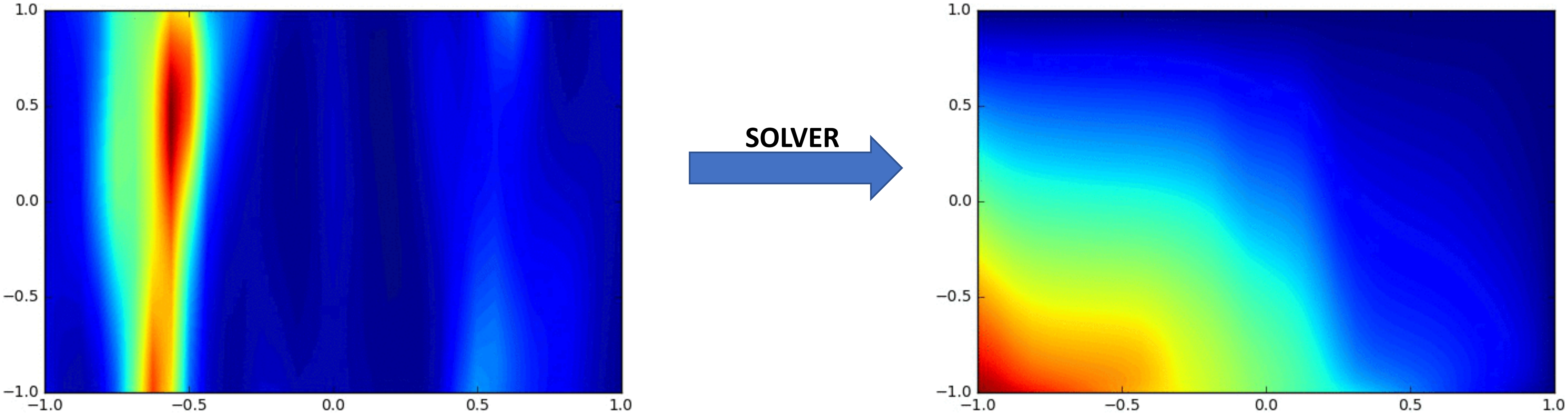


- where,

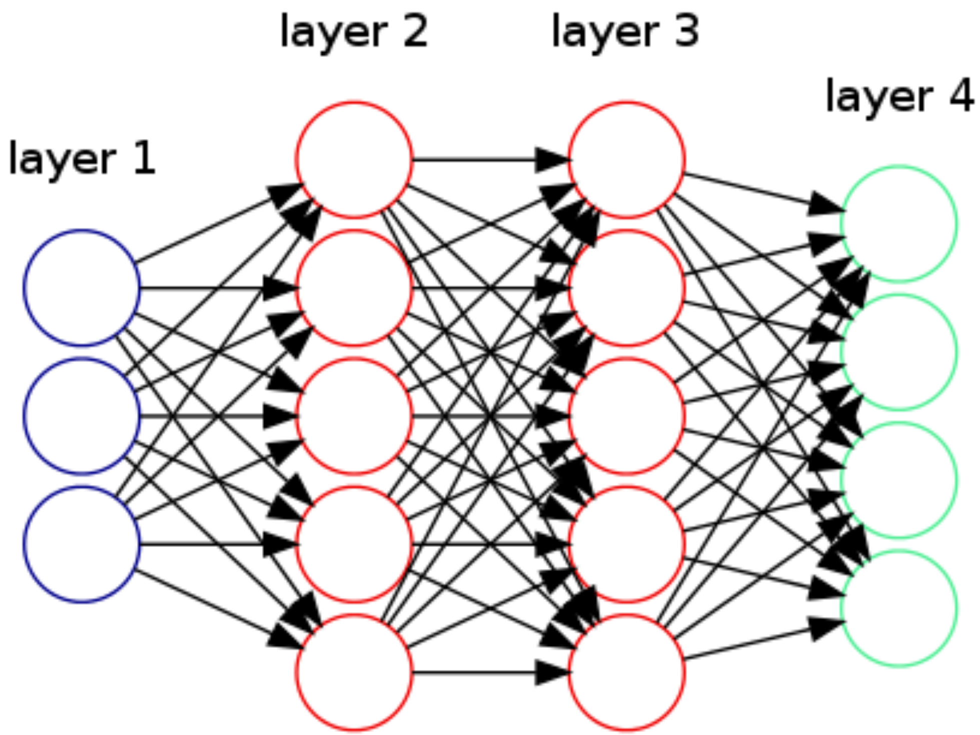
$$\mathbf{a} = \exp(g)$$
$$g \sim \mathcal{GP}(g(\mathbf{x}) \,|\, 0, k(\mathbf{x},\mathbf{x}'))$$

SE covariance

Lengthscales in the transformed space:
- 0.1 in the x-direction.
- 1.0 in the y-direction.


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
SOLVER
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

Deep Neural Network (DNN) surrogate



-Independent surrogate for each 'pixel' in the output.

$$\mathcal{F}(\mathbf{a};\theta^{(i)}):\mathbb{R}^{1089}\rightarrow\mathbb{R}$$

$$\theta=\{\mathbf{W}_l,\mathbf{b}_l:l\in\{1,2,\cdots,L,L+1\}\}$$

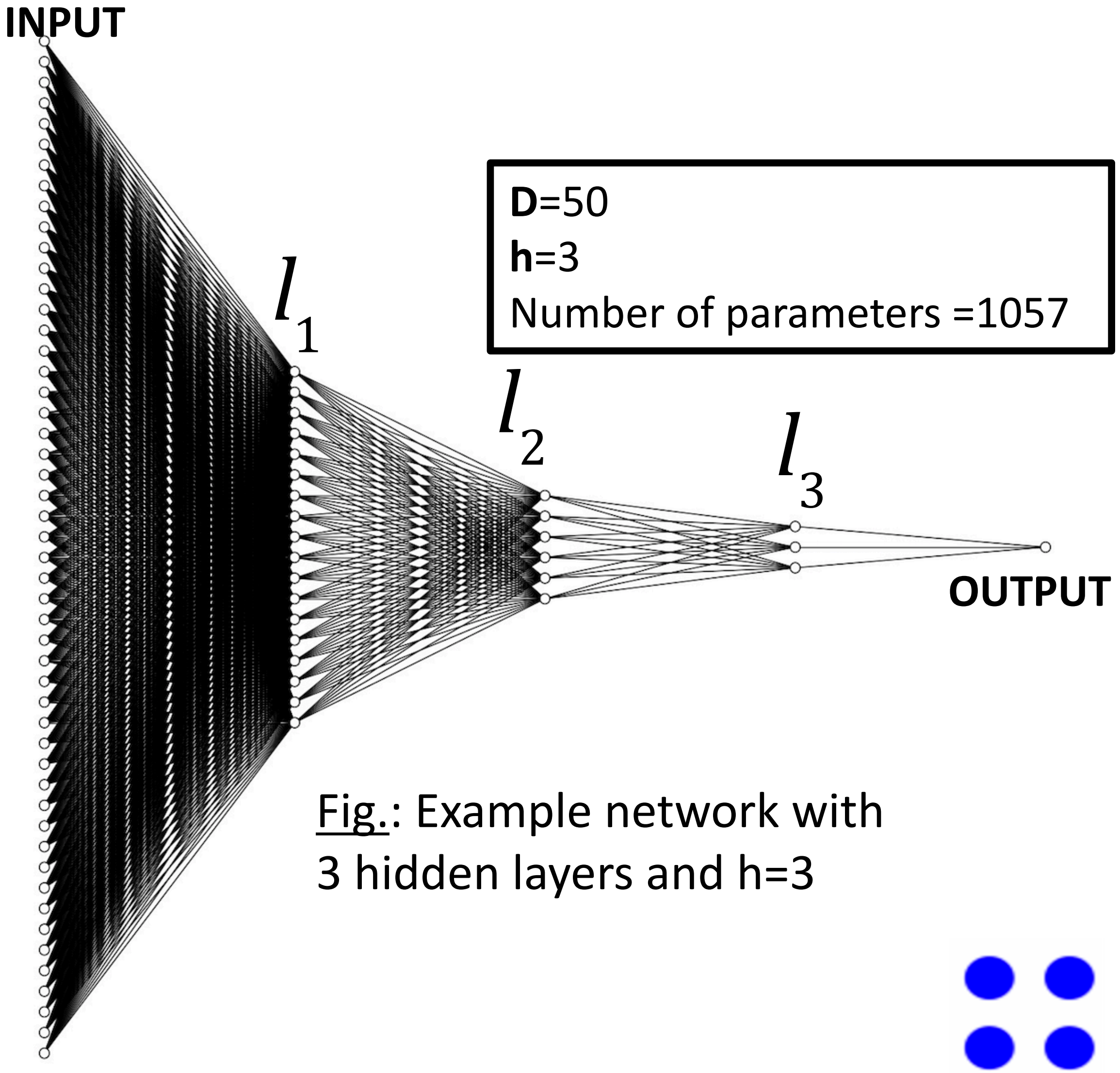
NETWORK ARCHITECTURE

$$\boxed{n_i = D\exp(\beta i)}$$

$$i \in \{1,2,\cdots,L\}$$

Why this way:

- Full network parameterized by just 2 numbers.
- Dimensionality reduction interpretation.



D=50
h=3
Number of parameters =1057

Fig.: Example network with 3 hidden layers and h=3

OPTIMIZATION

Likelihood model: $y \mid \mathbf{a},\theta,\sigma \sim \mathcal{N}(|\mathcal{F}(\mathbf{a},\theta),\sigma^2)$

$$\mathcal{L}_j(\theta,\lambda,\sigma;\mathbf{a}_j) = \log(\sigma) + \frac{1}{2\sigma^2}(y_j - \mathcal{F}(\mathbf{a}_j;\theta))^2$$

NEGATIVE LOG LIKELIHOOD

Full Loss function: $\mathcal{L} = \frac{1}{N}\sum_{j=1}^{N}\mathcal{L}_j + \lambda\sum_{l=1}^{L+1}\|\mathbf{W}_l\|^2$

REGULARIZER

$$\boxed{\theta^*,\sigma^*,\lambda^* = \arg\min \mathcal{L}}$$

ADAptive Moments (ADAM[1]) optimizer

- Backpropagation[2] to compute gradients.
- Use mini-batch size of 32.
- Initial stepsize set to $3\text{x}10^{-4}$.
- Drop stepsize by factor of 1/10 every 15k iterations.
- Train for 45k iterations.
- Moment decay rate: $\beta_1 = 0.9, \beta_2 = 0.999$.

$$\boxed{\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}}$$

References:

[1]- Kingma and Ba. *Adam: A method for stochastic optimization*. (2014).

[2]- Rummelhart and Yves, *Backpropagation: theory, architectures, and applications.* (1995).

PURDUE UNIVERSITY

SELECTING OTHER HYPERPARAMETERS

- Select number of layers L, width of final hidden layer, h and regularization parameter λ with cross validation.
- Perform cross-validation on one output point; reuse selected network configuration on all the remaining outputs.

$$\boxed{\mathbf{S}(\mathbf{a};\mathcal{F}) = \frac{1}{N_{val}} \sum_{i=1}^{N_{val}} \left(y_i^{true} - y_i^{pred}\right)^2}$$

PURDUE UNIVERSITY

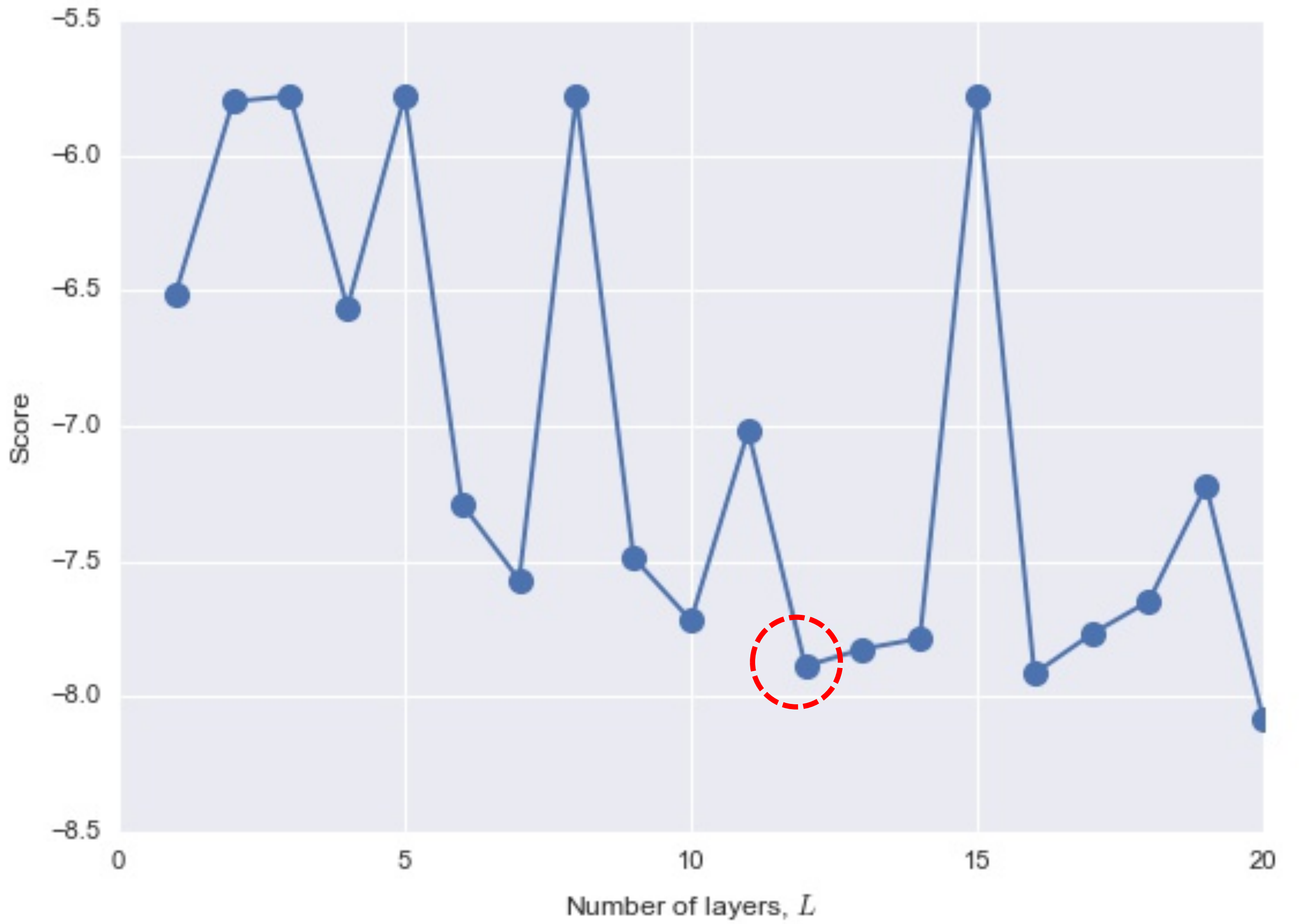


Fig.: Score vs number of layers

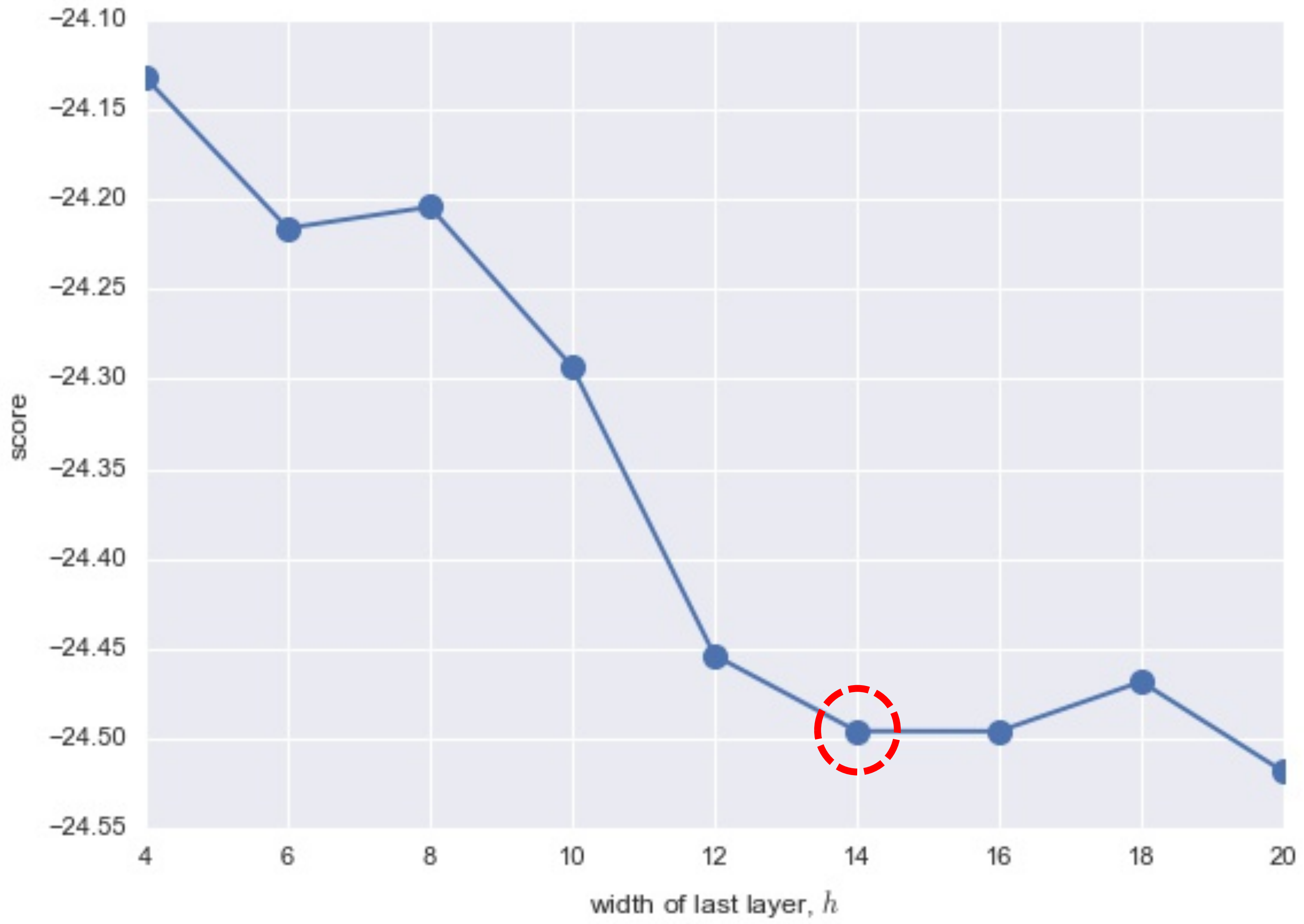


Fig.: Score vs width of last hidden layer

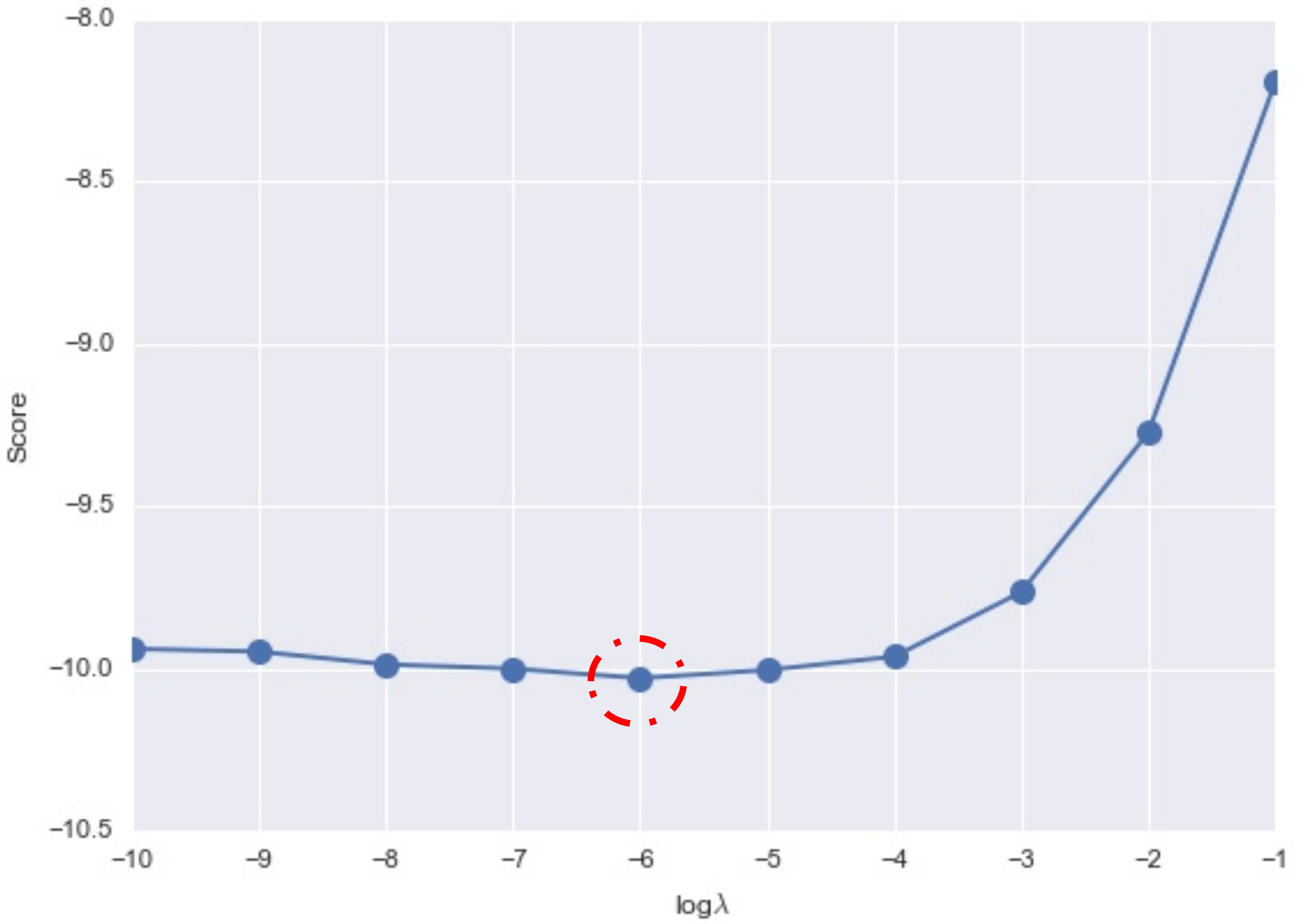


Fig.: Score vs log of weight decay

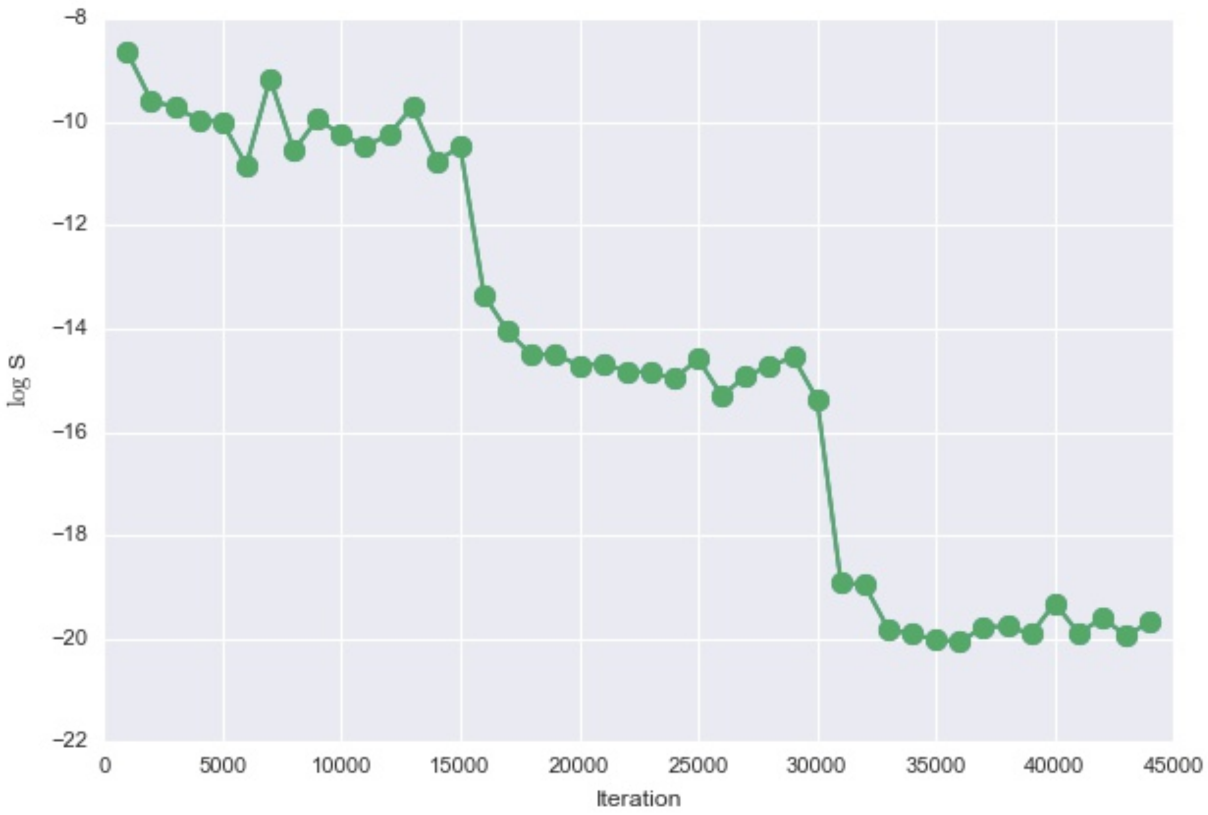


Fig.: Iteration vs Score

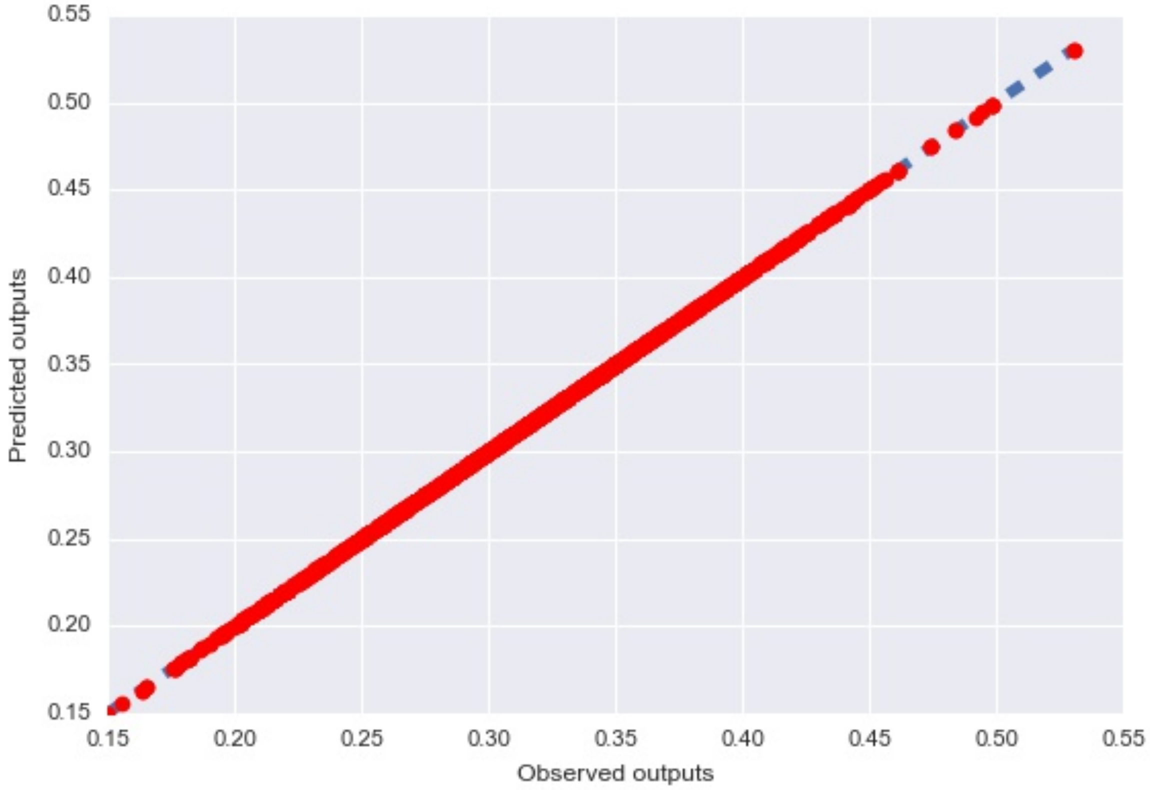
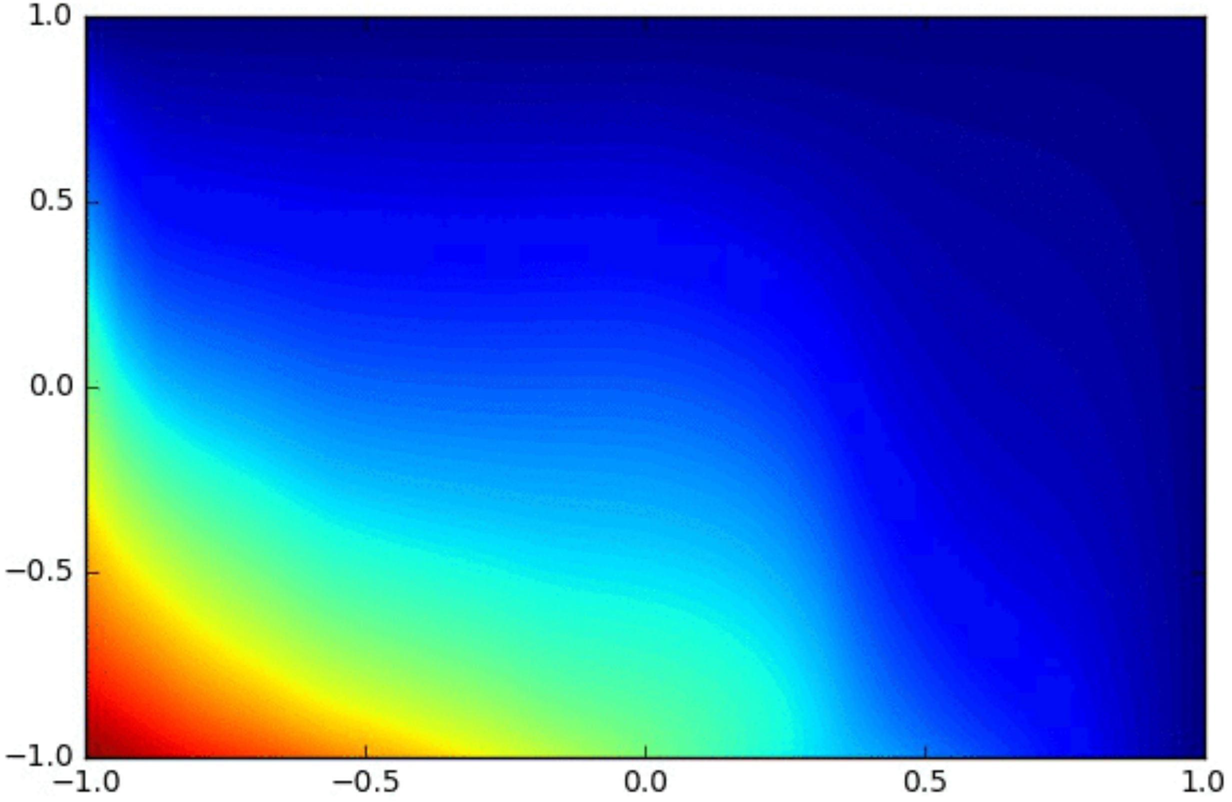


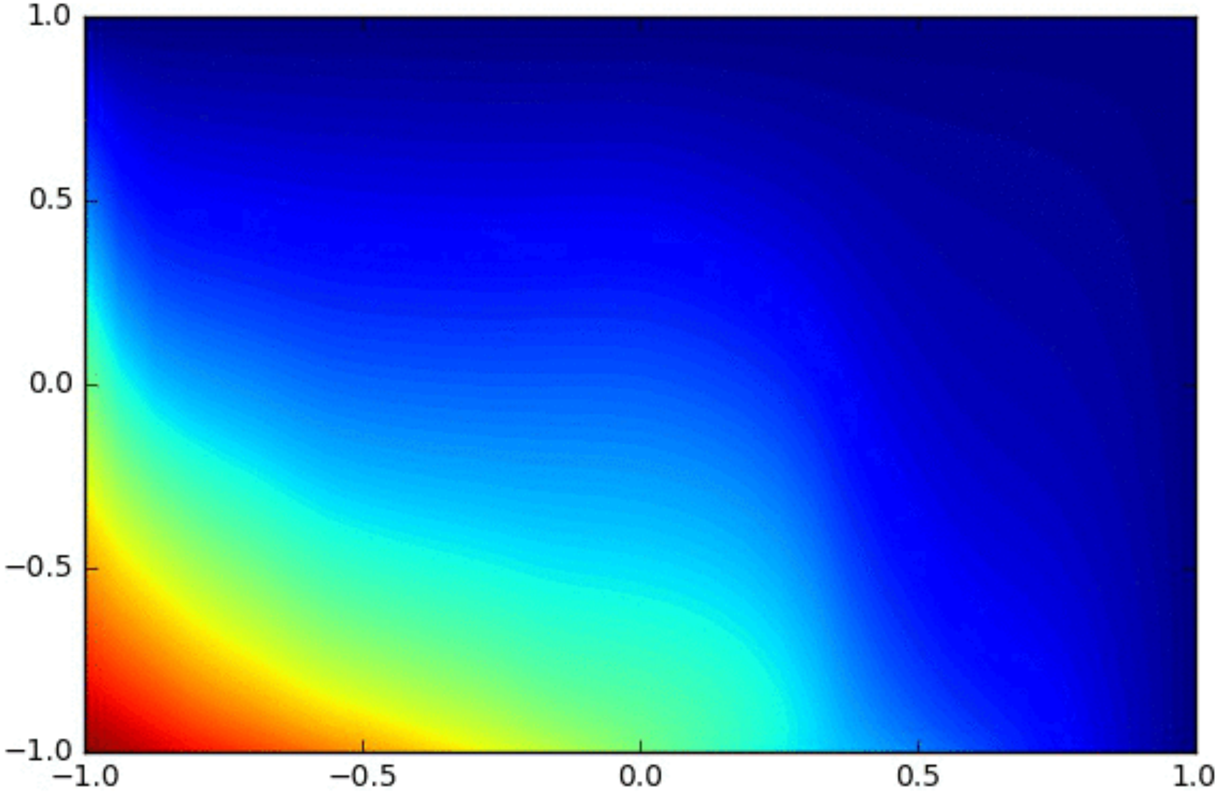
Fig.: Observed outputs vs predicted outputs on the test dataset.

Prediction of full solution

Predicted solution



True solution

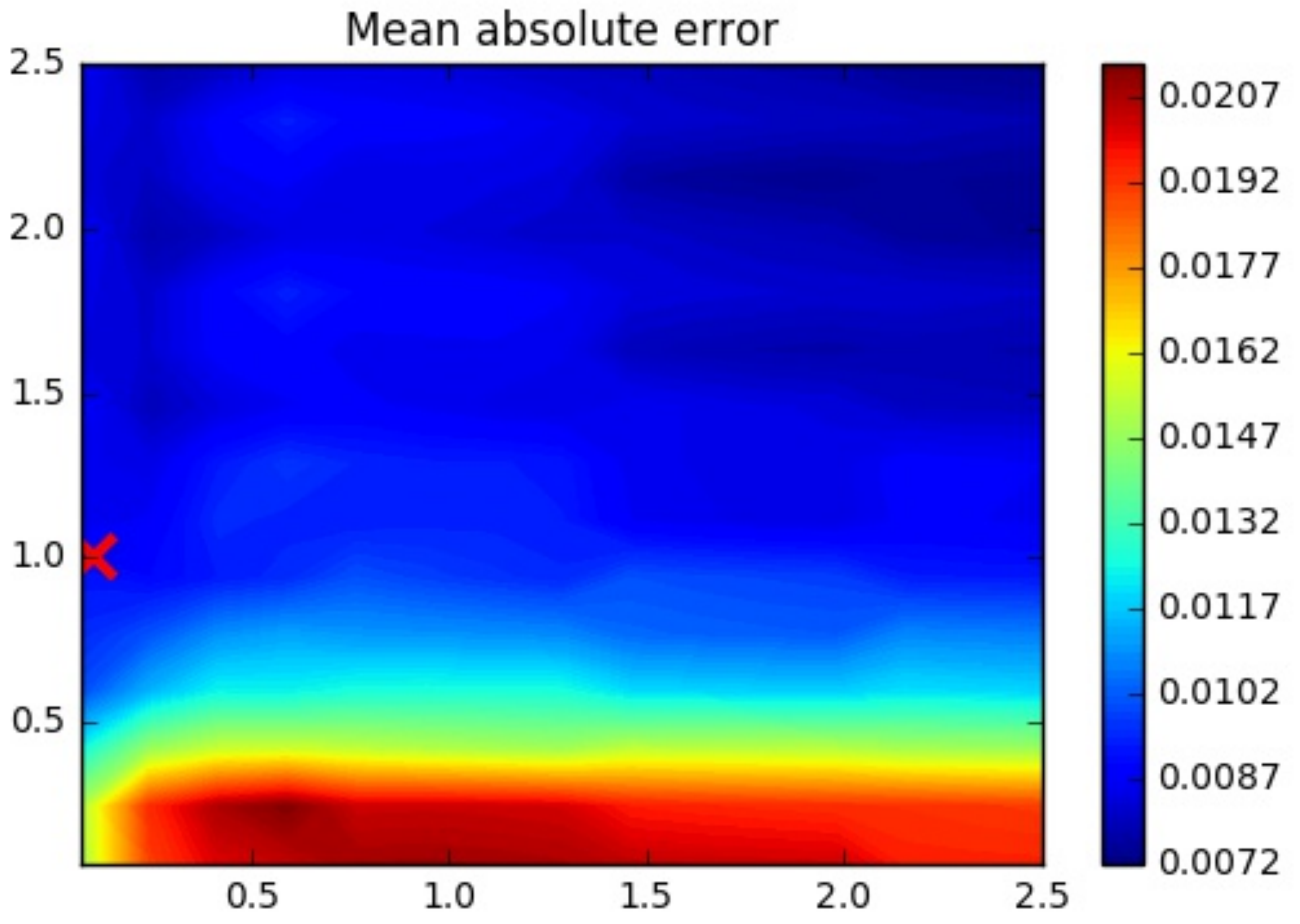


MSE (x10^{-6}): 2.56480241403

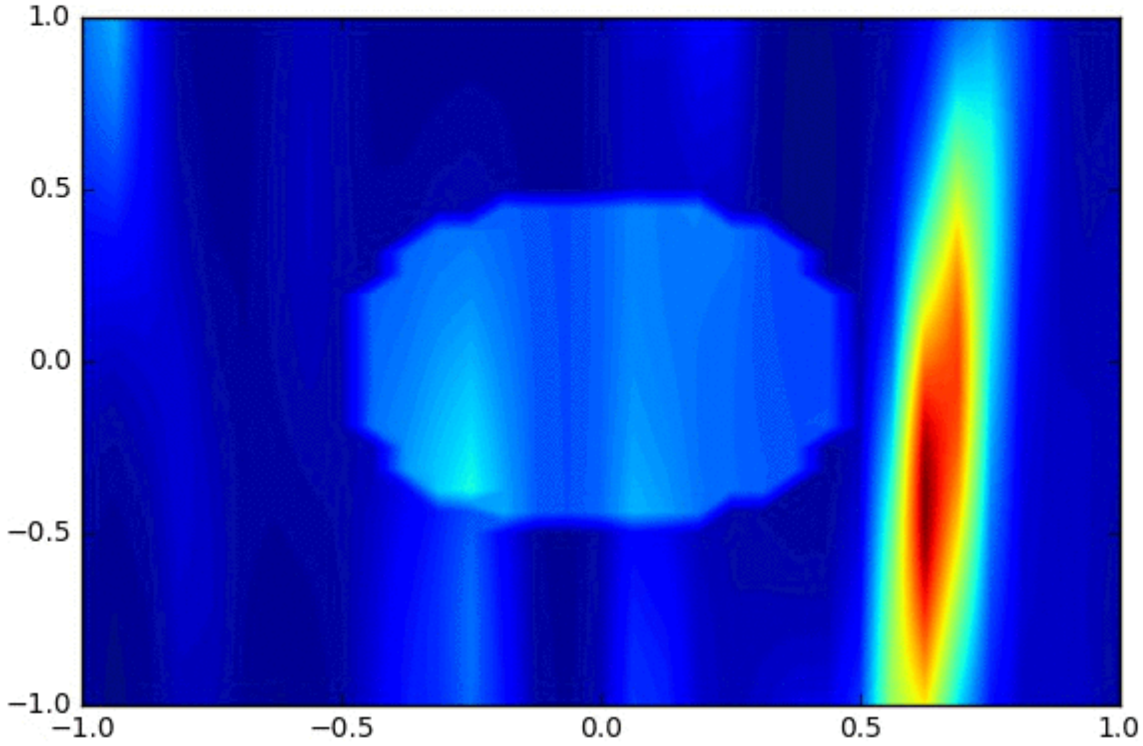
PURDUE UNIVERSITY

What if we predict solution on inputs from random fields that have different lengthscales ?

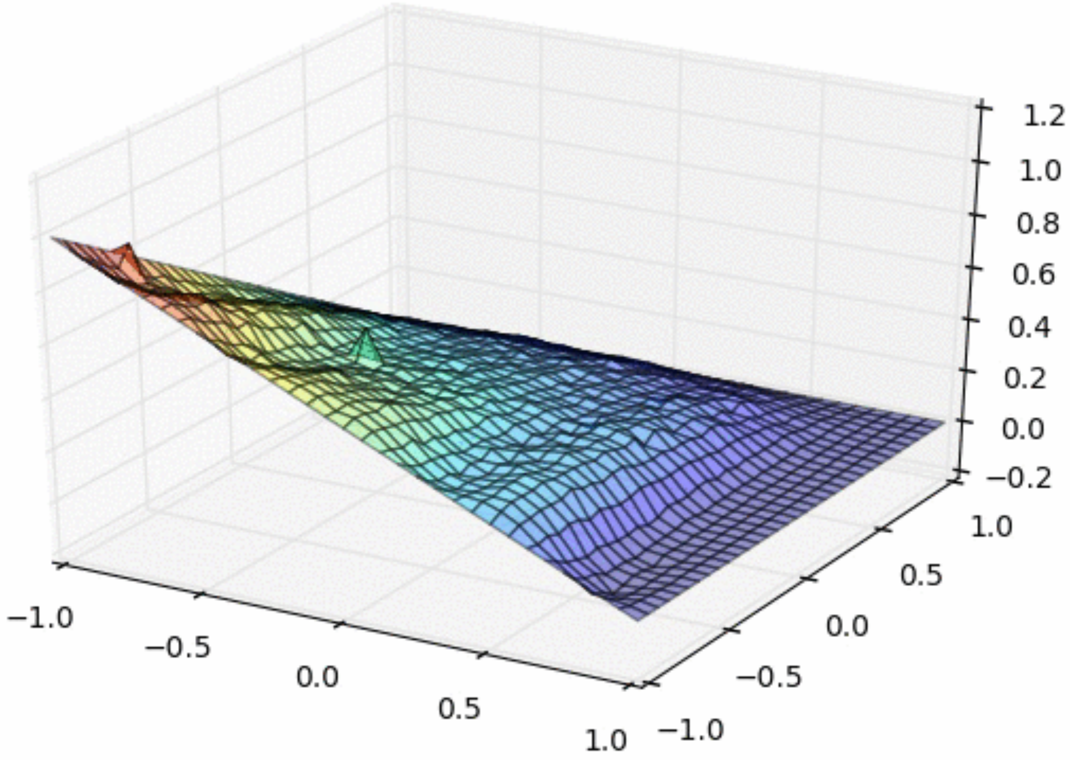
TESTING THE SURROGATE WITH DIFFERENT RANDOM FIELDS



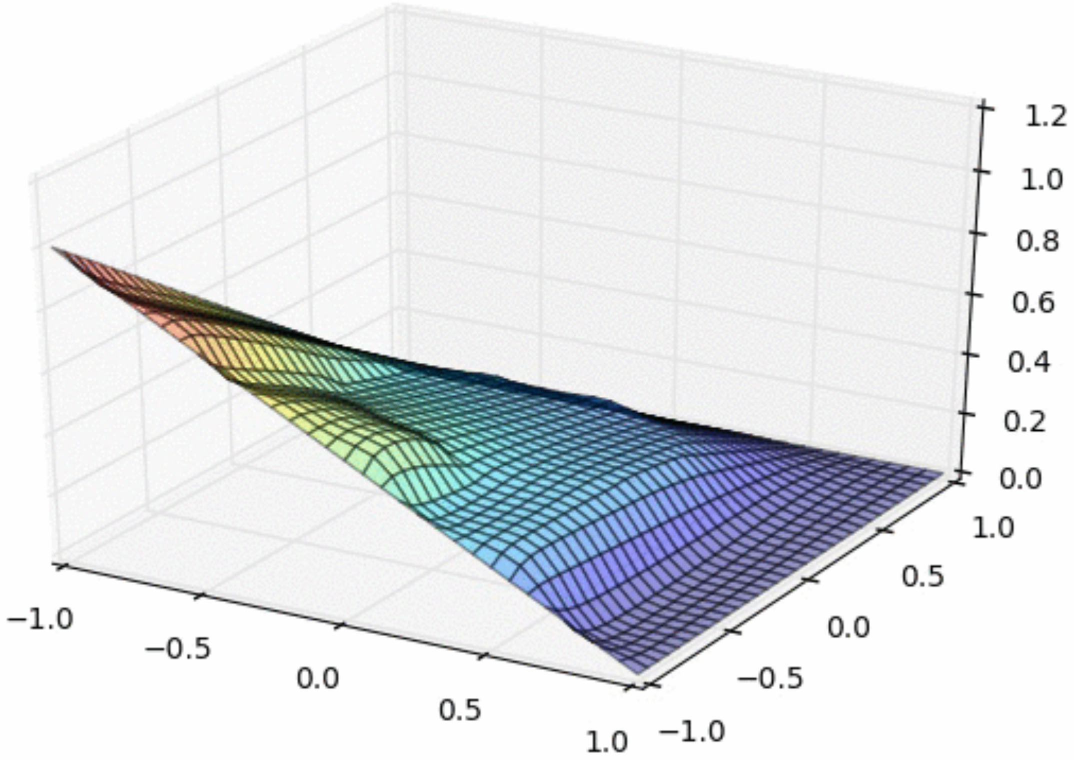
What about fields with discontinuities?



PREDICTED SOLUTION



TRUE SOLUTION



FUTURE DIRECTIONS

- Reduce data with unsupervised pretraining.
- Correlations between outputs (Multi-task learning).
- Fields with arbitrary spatial discretization (Fully convolutional networks).
- Bayesian training (stochastic variational inference).