Uncertainty Robust GRU-based Bayesian Approximation

1st UNIST-POSTECH-KAIST Datascience Contest

Team 월넛쿠키
김민우
이중원
연제혁

Keywords:

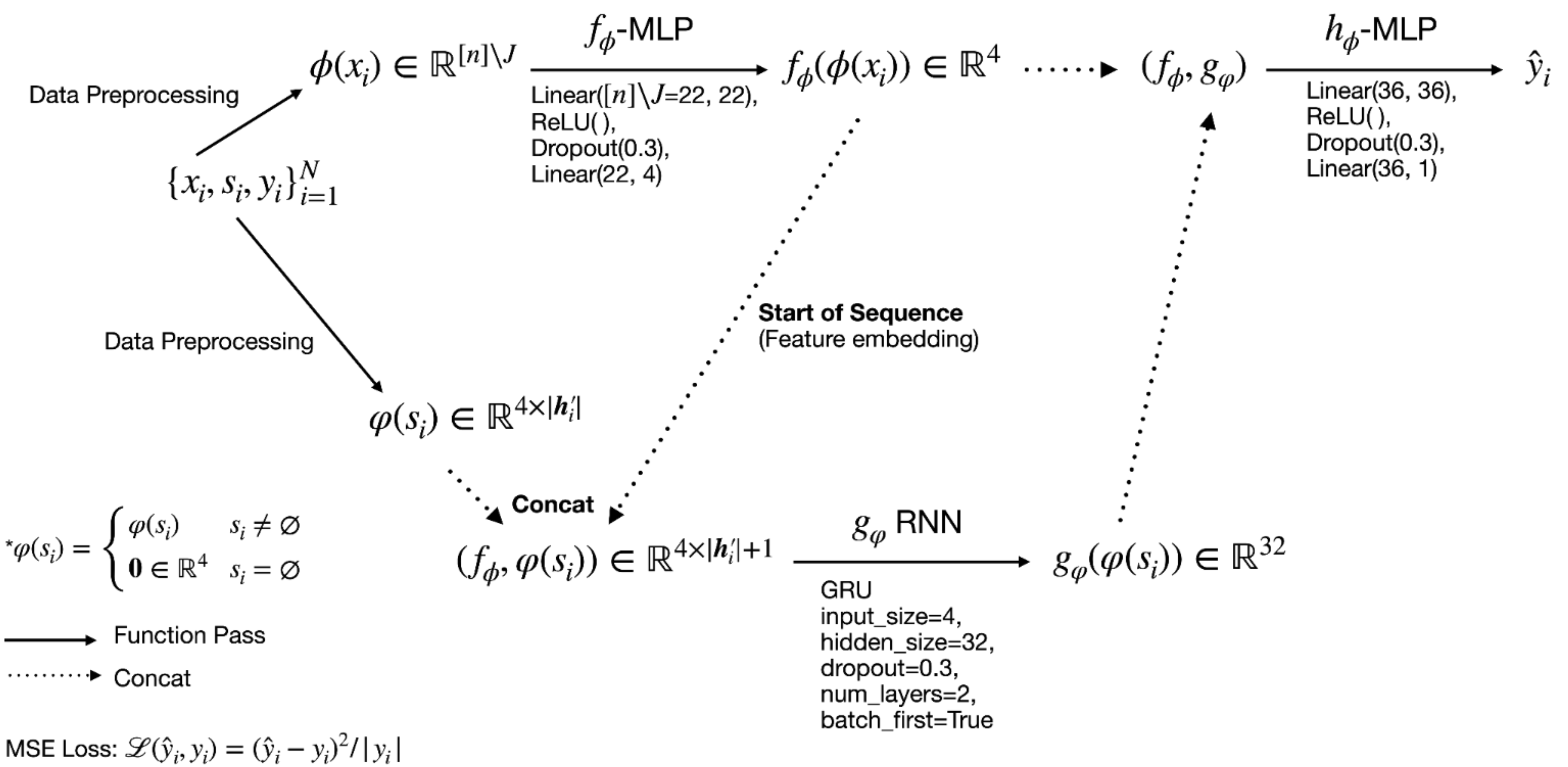
Denoising Regressor
Uncertainty Robust Knapsack Solver
Bayesian Approximation

Code Available

https://github.com/MinuKim-KAIST/datascience-contest.git

Introduction:
Sketch of the Presentation

Deep Learning Based Model



Advantages of the Model In a Nutshell

Key Question:
Why Deep Learning, and how can we cope with the lack of data?

01

Single Model.

Applicable to both tasks with a single model

02

Uncertainty Robust.

Risk management with various ML techniques, that are mathematically proved

03

Extendable.

Meta-learnable to different tasks, and capable of incorporating more and complex features

Our Results in a Nutshell

Our average validation MSE loss over 50 different random states are:

Task 1.
31977.8137

Task 2.
6239.97285

That is, for instance in Task 2, we are making predictions with average difference approximately 6239.
Our best SMAPE score provided by the competition organizers is:

SMAPE
20.27%
2nd Place(2021.12.19.)

Part I.
Data Preprocessing

Overview of the Data Preprocessing Techniques

Deep Learning:
Universal Approximator requires a minimum data preprocessing step

	Technique	Description
1	**Remove String Features**	Trade-off between information loss and parameter number reduce expected
2	**Normalize Data**	Model learns better when features are all normalized to have similar scales
3	**Sequence Embedding**	Robust representation of the sequential time series data are required
4	**Impute missing values**	Every information is valuable, especially when there is not enough data
5	**Gaussian Noise Augmentation**	Randomly adding Gaussian noise (in training phase) is equivalent to data augmentation

Learnable Parameter Analysis

LSTM with padding

15279

LSTM with sequence embedding done as in previous slide, without all string features

GRU with padding

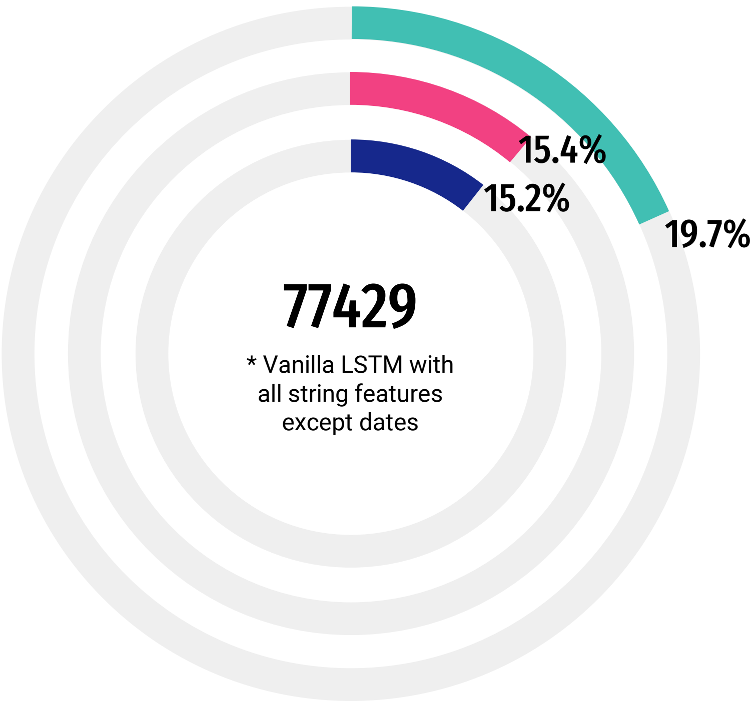
11951

The setting is the same with the above model, except replacing LSTM with GRU

GRU with no padding

11759

The same GRU model without all string features, but except the sequence embedding



Parameter numbers based on the model uploaded on the Github code, by running the following command:

```
sum(p.numel() for p in model_sequence.parameters() if p.requires_grad)
```

Sequence Embedding Techniques

Production rate rather than simple production values is important

Production quality is **continuous** before and after the **months of break**

Data normalized to ratio values as GAS/HRS and CND/HRS

Each row is normalized to take values in 0 ~ 1

GAS	61848	49026	0	0	0	43587	0	38925	33183	29958	...
CND	1650	1554	0	0	0	1365	0	1284	1269	1119	...
HRS	742	720	0	0	0	742	0	741	744	465	...

GAS	1.0027	0.8191				0.7066		0.6319	0.5365	0.7750	...
CND	1.0027	0.9732		Remove		0.8295	Remove	0.7813	0.7691	1.0851	...
HRS	0.742	0.720				0.742		0.741	0.744	0.465	...
Rest	0	0				**3**		**1**	0	0	

Takes the number of 0's that are removed

Part II.
Model Formulation

Overcoming Lack of Data and the Uncertainty

Lack of Data

Less than 300 data available, let alone the loss of data due to validation

Parameter Reducing

Reducing the number of parameters is critical as there are few data available

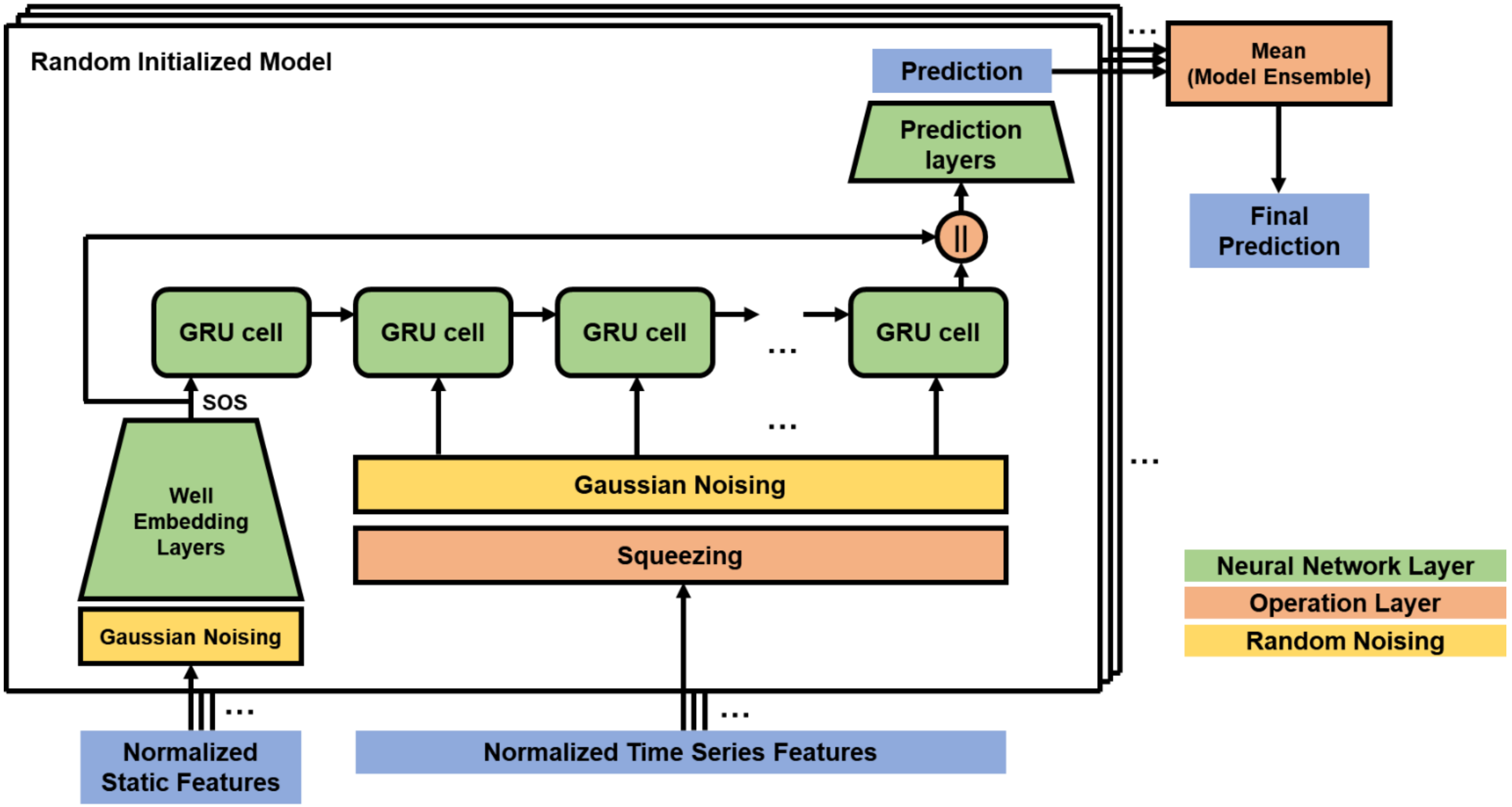
Adding Gaussian Noise

Adding Gaussian noise to data during the training phase serves as augmentation of unseen data

Bayesian Predictive Distribution

More than single predictions can be made with a same model, which better resolves the uncertainty of the model prediction

Overall Model Architecture



Part III.
Combinatorial Optimization

Existence of Unknown Features

Unknown Features = Possible Uncertainty

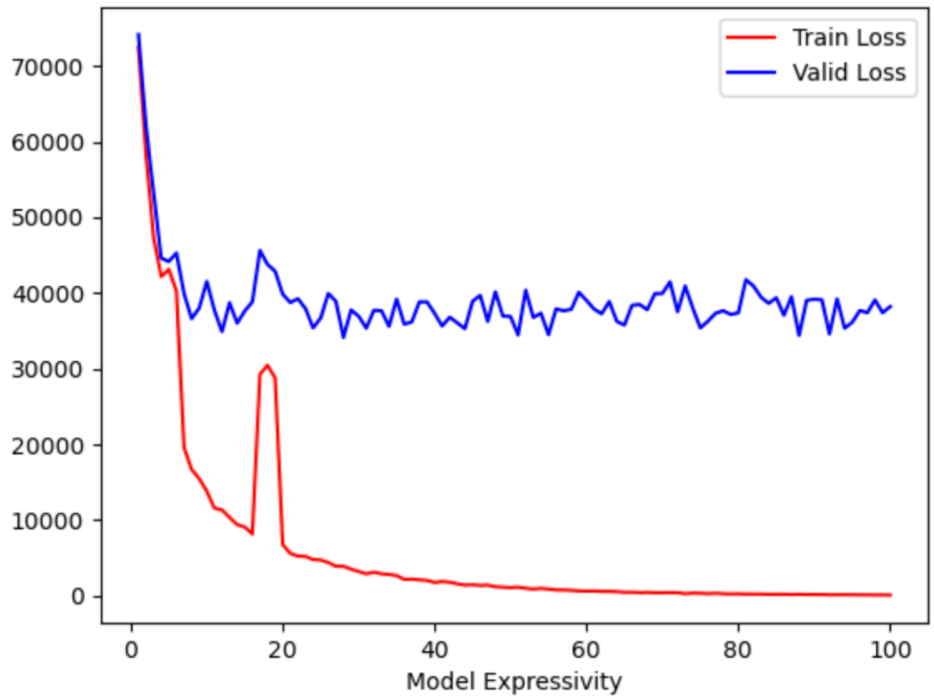


Figure 1. Train and Valid loss plot with universal approximators (MLP and/or XGBoost) with arbitrary model expressivity, averaged with 10 iterations under different random states. Valid loss does not drop below MSE = 30000, and hence further generalization is impossible.

Bayesian Predictive Distribution

Pred. Distribution = Risk Management

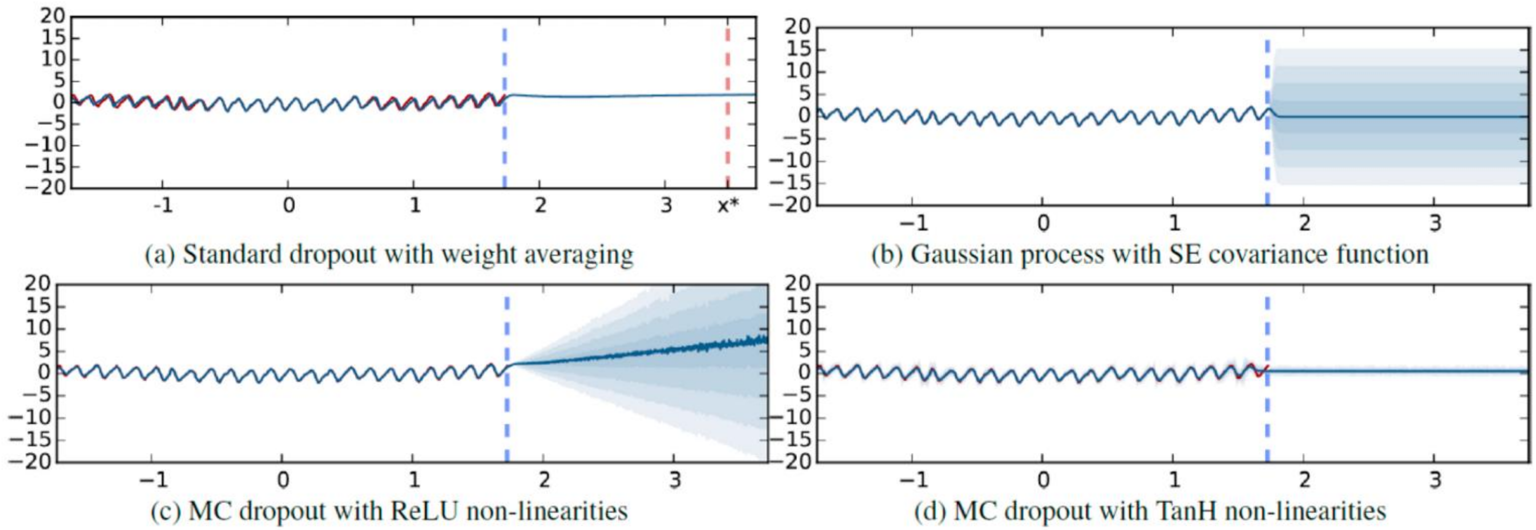


Figure 2. [Gal et al., 2016] Neural network with MC dropout and ReLU non-linearities generates Bayesian predictive distribution which serves as an uncertainty measure of the neural network. (a)-(d) compares predictive distributions of different models and layers.

Gal, Yarin, and Zoubin Ghahramani. (2016). International conference on machine learning. 1050-1059

Uncertainty Robust Knapsack Solver

$$\begin{aligned} \max \quad & \mathbb{E}_{\gamma}\left(\sum_i PROFIT\ (\hat{y}_{i,\gamma} \cdot x_i) \right) \\ \text{s.t.} \quad & \sum_i COST_i \cdot x_i \leq BUDGET \\ & x_i \in \{0,1\} \end{aligned}$$

Expectation-Maximization Model

$$\begin{aligned} \max \quad & v \\ \text{s.t.} \quad & \sum_i PROFIT\ (x_i \cdot \Phi_i^{-1}(1-\alpha)) \geq v \\ & \sum_i COST_i \cdot x_i \leq BUDGET \\ & x_i \in \{0,1\} \end{aligned}$$

Chance-contrained Model

Both of them are provided in our code:
https://github.com/MinuKim-KAIST/datascience-contest.git

Peng, Jin, and Bo Zhang. "Knapsack problem with uncertain weights and values." (2012).

Part IV.
Further Discussions

Possible Improvements

Data Preparation

More data can be collected, both in terms of the number and the features

Additional Feature Data

Static features such as geological data can be further incorporated in the model

Additional Sequence Data

Human decisions such as resting periods should contain more information; environmental data such as temp.

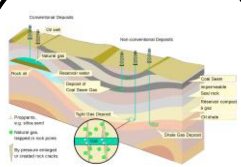
Domain Knowledge Informed DL

Rule-based classical prediction combined methods can be used, and geological information can be incorporated

Incorporating Geometric Data

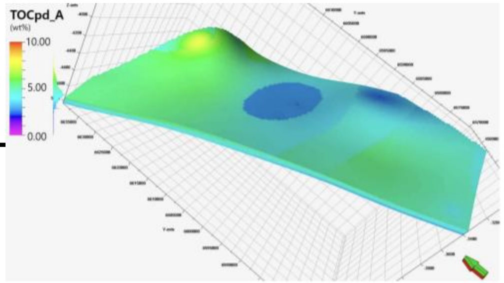
Remark

Shale gas production is more dependent on the reserves, than the properties of shale gas well itself. Hence, classical prediction utilizes geological survey.



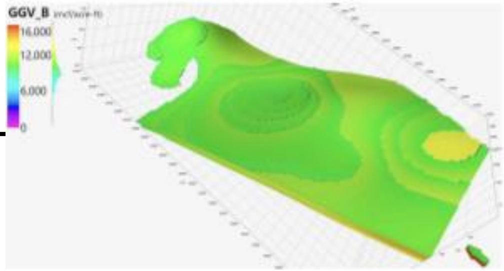
TOC (Total Organic Carbon)

Amount of carbon found in an organic compound and is often used as a non-specific indicator of water quality or cleanliness



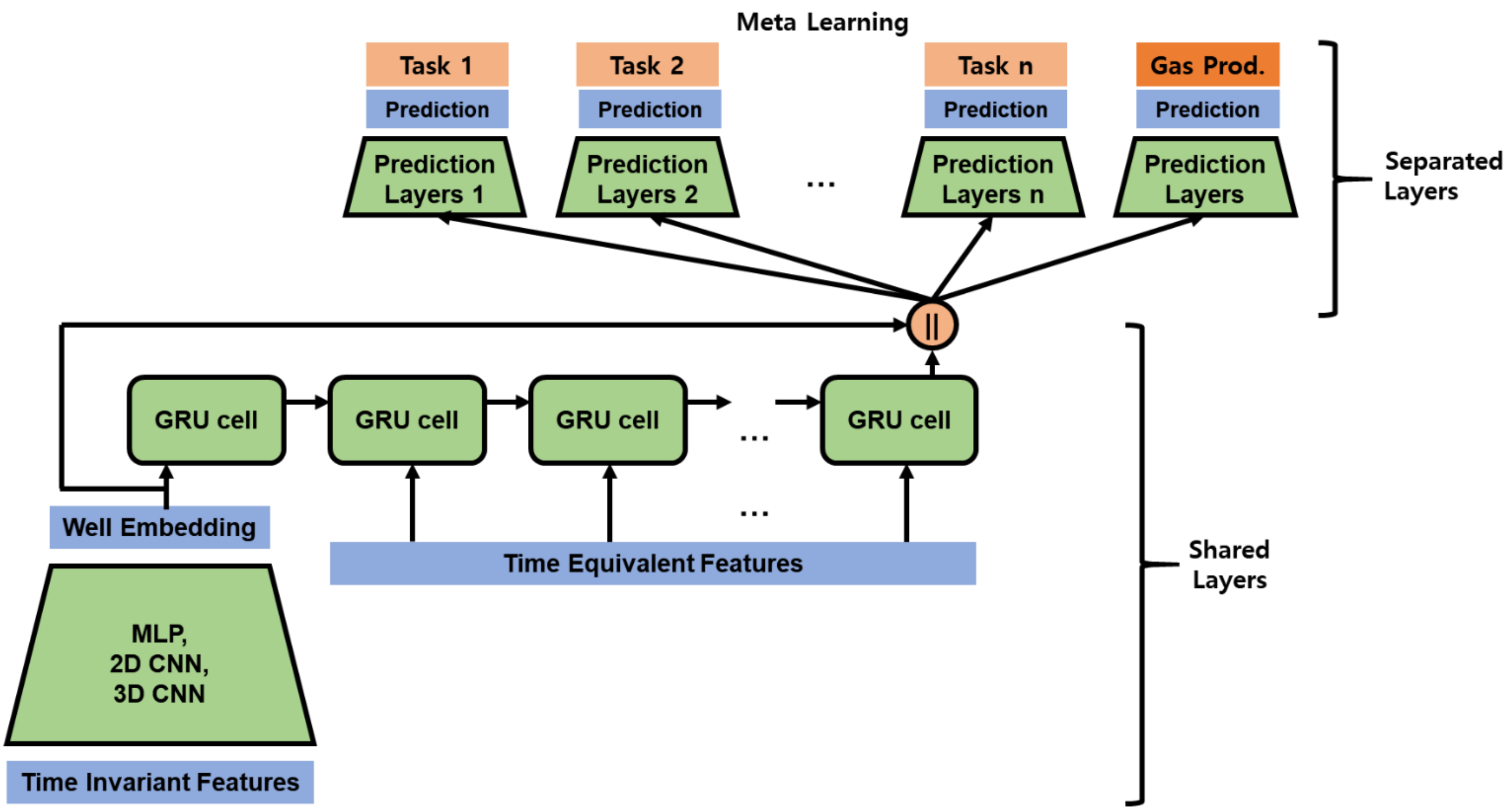
HIo (Original Hydrogen Index)

Represents the amount of hydrogen relative to the amount of organic carbon present in a sample.

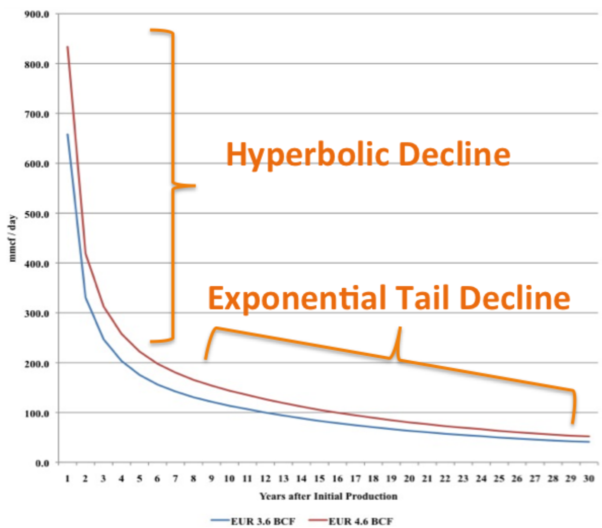


Deep Learning can be Extended Easily

Train for multi tasks (gas production, cnd production, well faults, ext.)

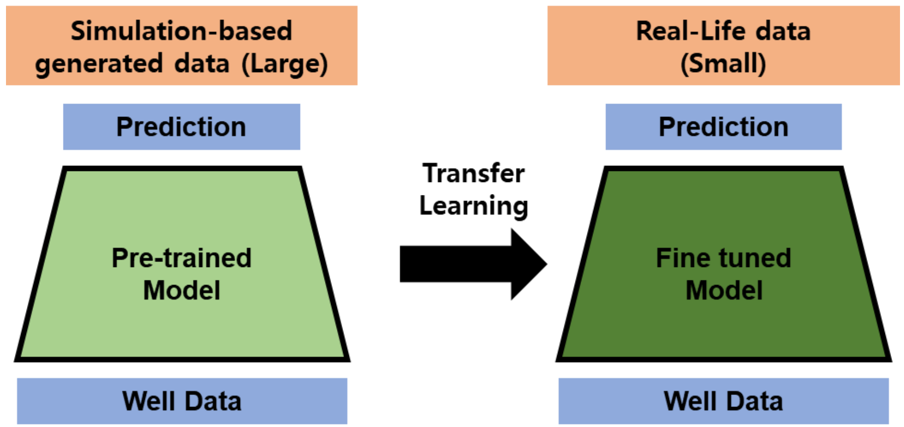


Inducing Domain Knowledge



$$P_t = IP \times \frac{1}{(1 + bDt)^{1/b}}$$

$$P_t = IPe^{-D_s t}$$



Wang, H. (2017). *Spe Journal*, *22*(02), 562-581.

References

[1] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).

[2] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

[3] Yazidi, A., et al. "Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?, Front." *Artif. Intell* 3.40: 10-3389.

[4] Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *international conference on machine learning*. PMLR, 2016.

[5] Rusak, Evgenia, et al. "A simple way to make neural networks robust against diverse image corruptions." *European Conference on Computer Vision*. Springer, Cham, 2020.

[6] Peng, Jin, and Bo Zhang. "Knapsack problem with uncertain weights and values." (2012).

[7] Lee, Kyungbook, et al. "Prediction of shale-gas production at duvernay formation using deep-learning algorithm." *SPE Journal* 24.06 (2019): 2423-2437.

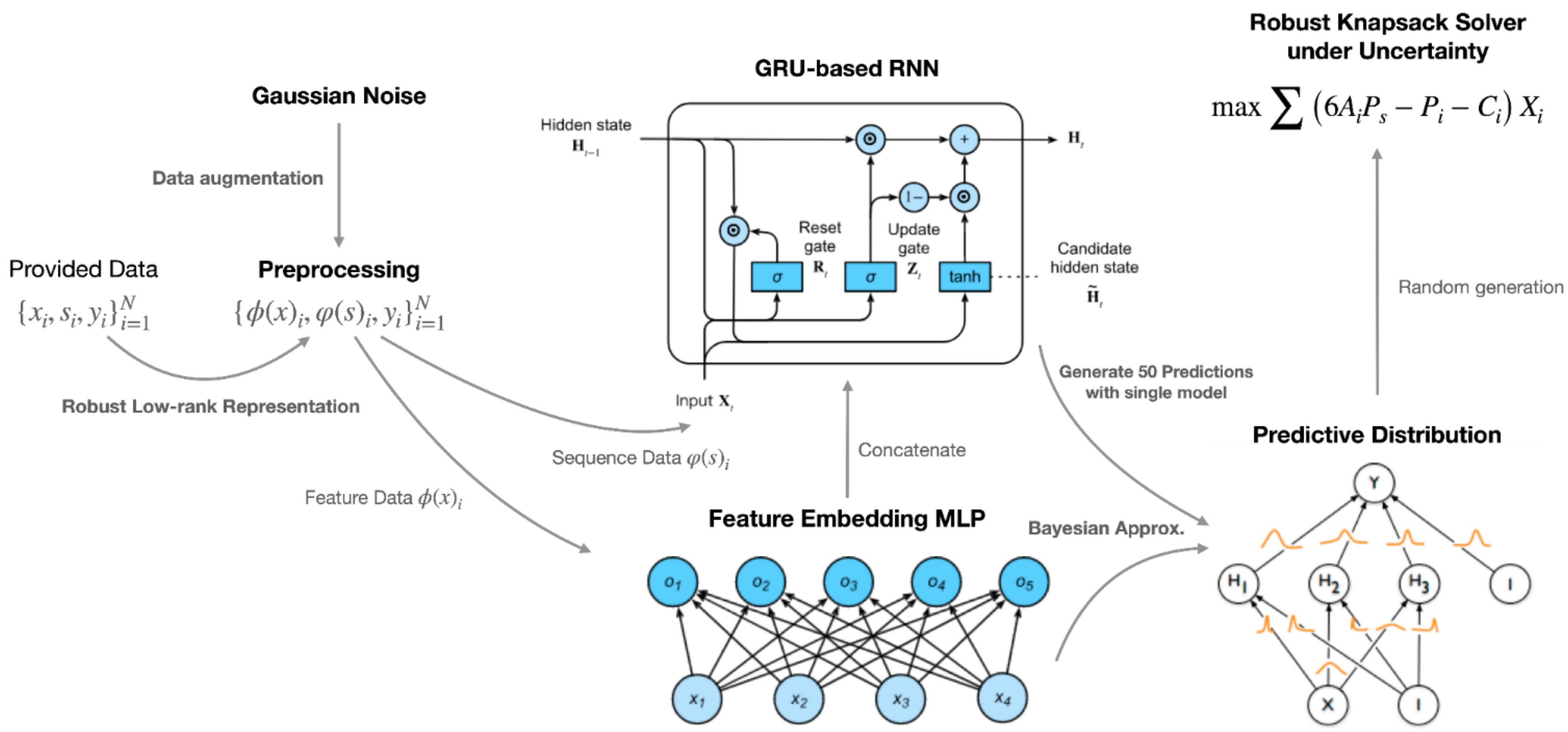
[8] Clark, Corrie, et al. "Hydraulic fracturing and shale gas production: technology, impacts, and policy." *Argonne National Laboratory* (2012): 1-16.

[9] Choi, Junhyung, et al. "A study on increase the productivity optimization solution using characteristics of geomechanical property in Western Canada Shale Basin." *Journal of Petroleum and Sedimentary Geology* 2.1 (2020): 24-35.

[10] Wang, HanYi. "What factors control shale-gas production and production-decline trend in fractured systems: a comprehensive analysis and investigation." *Spe Journal* 22.02 (2017): 562-581.

Appendix

Overview of the Proposed Model and Techniques



Technique Overview

Lack of Available Dataset

Uncertainty Robustness

- Generate **Bayesian Predictive Distribution** with random activation of Dropout layers
- **Uncertainty robust Knapsack** Solver using the generated distribution

Few-shot Learning

- **Imputing missing data** to maximize utility of the available data
- **Gaussian noise** addition during training process to emulate data augmentation
- **Reducing the number of parameters** of the universal approximators

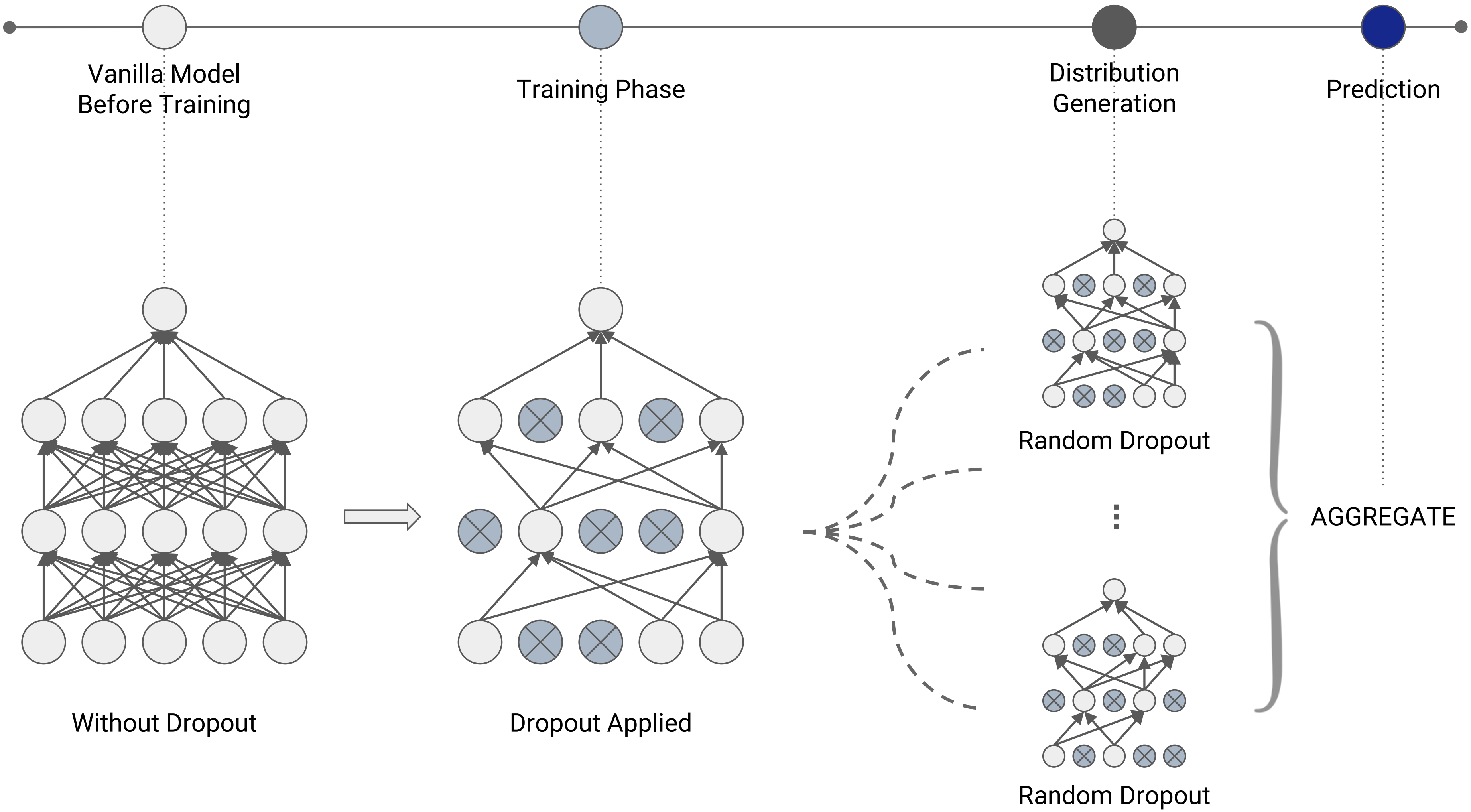
Removing String Features reduces Parameters

Stimulation Fluid: Oil: CWS-DynaGel, Water: SLB-WaterFrac (WF), Slickwater, ... (TOTAL **20**)
On Prod YYYY/MM/DD: 2015.5.21, 2014.10.16, ... (TOTAL **154**)
First Prod YYYY/MM (TOTAL **69**),
Last Prod YYYY/MM (TOTAL **3**),
Proppant Composition: Ceramic/Sand, Sand (TOTAL **2**)
Proppant Name 1: FTecISP, HydroProp, Sand, ... (TOTAL **13**)
Proppant Size 1: 40/70, 40/80, 30/50, ... (TOTAL **6**)

⋮

e.g. using "Stimulation Fluid" feature requires using a parameter of dimension 20. Trade-off between information loss and parameter size.

Bayesian Predictive Distribution



Gal, Yarin, and Zoubin Ghahramani. (2016). International conference on machine learning. 1050-1059

Fine Tuning of the Hyper-parameters

hyperparameters	searched space
well embedding dim	4 8 **16** 32 64
GRU hidden dim	8 16 32 **64** 128
GRU num layer	1 **2** 3 4
dropout	0.1 0.2 **0.3** 0.4 0.5
feature noise	0.01 0.03 0.05 **0.1** 0.3 0.5
sequence noise	0.01 0.03 0.05 **0.1** 0.3 0.5
batch size	2 4 8 **16** 32 64
Xavier initialization	**True** False
activation function	ReLU **LeakyReLU** SiLU