Input Decay

Yuze Zhou

Feb.2022

Cost Function and Regularization

- By denoting the function computed by MLPs as $f(x;\theta)$ with corresponding label y and loss function l, as well as the input decay term $C_{ID}(\theta)$, the cost function is:

$$C = \frac{1}{2N}\sum_{i=1}^{N} l(y_i, f(x_i;\theta)) + C_{ID}(\theta)$$

- Let $\theta^{(i)}$ be the parameter on the ith layer, and $\theta^{(1)}_{jh}$ be the first layer weight linking input j to hidden layer h, the input decay regularization for the jth input is therefore:

$$C^{j}_{ID}(\theta) = \sum_{h=1}^{H}(\theta^{(1)}_{jh})^2 = \|\theta^{(1)}_{j}\|^2_2$$

- The input decay penalty is therefore formulated as:

$$C_{ID}(\theta) = \phi\sum_{h}\frac{C^{j}_{ID}(\theta)}{\eta + C^{j}_{ID}(\theta)}$$

Cost Function and Regularization

- The function of $y = \frac{x^2}{\eta + x^2}$ again x with $\eta = 1$ is shown below:

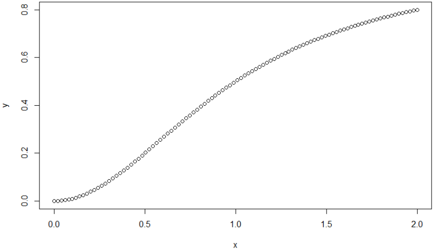

Figure 1: Regularization Function

Model Description

- **Hidden Layer:** Three hidden layers are incorporated into the model, each with size 600, 300 and 200.
- **Activation Function:** For each hidden layer, the activation function is sigmoid function.
- **Loss Function:** The loss function l used here is the mse loss $l(y_i, f(x_i;\theta)) = (y_i - f(x_i;\theta))^2$
- **Benchmark Evaluation** The benchmark to evaluate the model is the **correlation** between the predicted value $\hat{y}$ and the true label y.
- **Training set and Validation Set:** The training set and the validation set were splitted into proportion $5/7$ and $2/7$

Benchmark Model

- Two benchmarks were used for comparison: Linear Regression and Neural Nets without any regularizations.

Model	Training Correlation	Validation Correlation
Linear Regression	0.1416	0.0701
Neural Nets	0.1201	0.0752

Parameter Tuning and Hyper-parameter Selection

- For input decay, the value of η is chosen globally as 1, with ϕ being the regularization parameter that needs to be tuned.
- The optimizer for each model is chosen as **Adam**; the global learning rate is $1e-3$ and, the hyper-parameters are set as $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-6$.

ϕ	best epoch	training correlation	validation correlation	training mse	validation mse
1e-10	6	0.1167	0.0770	2.400e-3	3.239e-3
1e-9	7	0.1235	0.0782	2.410e-3	3.255e-3
1e-8	5	0.1246	0.0815	2.370e-3	3.212e-3
1e-7	7	0.1236	0.0807	2.378e-3	3.219e-3
1e-6	28	0.0893	0.0642	2.416e-3	3.252e-3
1e-5	28	0.0792	0.0572	2.418e-3	3.256e-3
1e-4	24	0.0654	0.0503	2.425e-3	3.255e-3
1e-3	30	0.0679	0.0451	2.438e-3	3.282e-3

- Selection of the tuning parameter ϕ is based on the validation correlation. The best candidate for the hyper-parameter ϕ is $1e-8$.

Feature Selection

- The baseline model for feature selection is LASSO, where the best regularization parameter λ is chosen via cross-validation.
- The summary of LASSO with 5-fold cross-validation is presented as followed:

signal features	noise features	training correlation	validation correlation
19/240	1/100	0.0573	0.0278

Feature Selection

- The feature selection rule is based upon thresholding $C^j_{ID}(\theta)$.
- Given a fixed level of threshold, if the l_2 norm for weights linking feature j in the inputs layer is larger than the threshold value, the feature would be selected; otherwise the feature is neglected.
- To evaluate the performance of feature selection, we have two different correlation benchmarks for comparison.

  Trained Weights: We used the weights from the previous trained neural nets, but only use the selected features to get predicted values for $\hat{y}$ for the validation set.

  Re-training: We re-trained the neural nets with the same layer structure, but only the selected features are used for the training set, and use the re-trained model to get predicted values for the validation set. The hyper-parameters of the optimizer for the re-trained net are the same as the trained one.

Feature Selection

- The number of features selected and the correlation after selection is presented in the following table:

threshold	signal features	noise features	correlation (trained)	correlation (re-train)
1.3	237/240	29/100	0.0813	0.0820
1.5	235/240	6/100	0.0812	0.0836
1.7	231/240	2/100	0.0814	0.0807

Conclusion

- **Input Decay** could suppress all the weights linking to features with less importance in the input layer, without harming the prediction results.
- The threshold value for feature selection needs to be picked manually.