Deep Learning

AlejandroMllo

This document serves as a very brief summary of the topics covered in each chapter of the book Deep Learning [1].

Disclaimer: This document is completely extracted from [1], the author does not attribute any ownership over the material.

7 Regularization for Deep Learning

- Regularization: strategies designed to reduce the test error, possibly at the expense of increased training error. Its goal is to make a model match the true data-generating processes. An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.
- Many regularization approaches are based on limiting the capacity of the models by adding a parameter norm penalty $\Omega(\boldsymbol{\theta})$ to the objective function J. The regularized objective function is:

  $$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, \infty)$ is a hyperparameter that weights the relative contribution of the norm penalty term, Ω, relative to the standard objective function J.
- For NN is typically used a parameter norm penalty Ω that penalizes only the weights of the affine transformation at each layer and leaves the biases unregularized.
- L^2 Parameter regularization (weight decay): this strategy drives the weight closer to the origin by adding the regularization term $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$ to the objective function.
- L^1 regularization: $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1 = \sum_i |w_i|$. It controls the strength of the regularization by scaling the penalty Ω using a positive hyperparameter α. The regularization contribution to the gradient does not scale linearly as in L^2. Its solution tends to be more sparse than L^2's solution.
- It is possible to think of a parameter norm penalty as imposing a constraint on the weights.
- It is possible to use explicit constraints rather than penalties.
- Weight decay will cause gradient descent to quit increasing the magnitude of the weights when the slope of the likelihood is equal to the weight decay coefficient.
- Data set augmentation: generate new $(\boldsymbol{x}, y)$ pairs by transforming the $\boldsymbol{x}$ inputs in the training set.
- NN prove not to be very robust to noise. One way to improve the robustness of NN is simply to train them with random noise applied to their inputs. This is also a form of data augmentation.
- When comparing ML benchmark results, taking the effect of dataset augmentation into account is important.
- Noise applied to the weights can be interpreted as equivalent to a more traditional form of regularization, encouraging stability of the function to be learned.
- Most datasets have some number of mistakes in the y labels. It can be harmful to maximize $\log p(y|\boldsymbol{x})$ when y is a mistake. One way to prevent this is to explicitly model the noise on the labels. For example, assume that for some small constant ϵ, the training set label y is correct with probability $1 - \epsilon$, and otherwise any of the other possible labels might be correct.
- In the paradigm of semi-supervised learning, both unlabeled examples from $P(\mathbf{x})$ and labeled examples from $P(\mathbf{x}, \mathbf{y})$ are used to estimate $P(\mathbf{y} \mid \mathbf{x})$ or predict $\mathbf{y}$ from $\mathbf{x}$. In deep learning, semi-supervised learning refers to learning a representation $\boldsymbol{h} = f(\boldsymbol{x})$. The goal is to learn a representation so that examples from the same class have similar representations.
- Multitask learning is a way to improve generalization by pooling the examples (which can be seen as soft constraints imposed on the parameters) arising out of several tasks.
- Among the factors that explain the variations observed in the data associated with different tasks, some are shared across two or more tasks.

- Epoch: a training iteration over the dataset.
- Early stopping: every time the error on the validation set improves, a copy of the model parameters is stored. When the training algorithm terminates, it returns these parameters, rather than the latest parameters. The algorithms terminates when no parameters have improved over the best recorded validation error for some pre-specified number of iterations. See algorithm 7.1 on the book.
- Parameter sharing: regularization method to force sets of parameters to be equal.
- Any model that has hidden units can be made sparse.
- Bootstrap aggregating (bagging) is a technique for reducing generalization error by training several different models separately, than have all models vote on the output for test examples. This is an example of a general strategy in ML called model averaging. Techniques employing this strategy are known as ensemble methods.
- Dropout trains an ensemble consisting of all subnetworks that can be constructed by removing nonoutput units from an underlying base network. When extremely few labeled training examples are available, dropout is less effective.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.