Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI 502] Extracting and Composing Robust Features with Denoising Autoencoders

1. Paper Summary

The difficulty of learning in deep architectures was resolved by the use of an unsupervised training criterion to perform a layer-by-layer initialization. This kind of initialization method yields a starting point, from which a global fine-tuning of the model's parameters is then performed using another training criterion appropriate for the task at hand. While the unsupervised learning of mapping that produces good intermediate representation to avoid getting stuck in the poor solution seems to be the key to this success, still little is understood for the explicit criteria that leads to good representation. Here, the author hypothesized and investigated an additional specific criterion which is robustness to partial destruction of the input and to verify the validity of its usefulness, a modification of the autoencoder framework is proposed to explicitly integrate the robustness to partially destroyed inputs.

The basic autoencoder takes an input x and deterministically maps it to a hidden representation $\mathrm{y} = \mathrm{f}_{\theta}(x) = s(Wx + b)$. Then, it is further mapped back to a reconstructed vector $\mathrm{z} = \mathrm{g}_{\theta'}(\mathrm{y}) = \mathrm{s}(\mathrm{W}'\mathrm{y} + \mathrm{b}')$ in input space where the weight matrix of the reverse mapping may optionally be constrained by $\mathrm{W}' = W^T$. Here, the denoising autoencoder tries to reconstruct a clean repaired input from a corrupted partially destroyed one. It is done by replacing certain portion of the initial input x to 0 or by adding gaussian random noise. But just like the basic autoencoder, the parameters are optimized to minimize the average reconstruction error between the uncorrupted input and the reconstructed input where the mean squared error or cross entropy is usually chosen.

Through the above way of model construction, the author expects to capture implicit invariances in the data so that interest features can be extracted. With the manifold learning perspective, the hypothesis can be further supported as the denoising autoencoder can be seen as a way to define and learn a manifold. Overall, the model learns a stochastic operator $\mathrm{q}(\mathrm{X}|\hat{X})$ which leads the low probability corrupted inputs $\hat{X}$ to high probability uncorrupted inputs X which generally locate on or near the manifold. Also, with generative model perspective, it can be derived that minimizing the cross-entropy in the denoising autoencoder is equivalent to maximizing a variational bound on a particular generative model $\mathrm{p}(\mathrm{X}, \hat{X}, Y)$. Here, $\mathrm{q}^0(\mathrm{X}, \hat{X}, Y) = \mathrm{q}^0(X) q_D(\hat{X}|X) \delta_{f_\theta(\hat{X})}(Y)$ is used as an auxiliary model in the context of a variational approximation of the log-likelihood of $\mathrm{p}(\hat{X})$ where $\mathrm{q}^0(X)$ denotes the empirical distribution associated to n training inputs and $\delta_{f_\theta(\hat{X})}(Y)$ puts mass 0 when $f_\theta(\hat{X}) \neq Y$.

The experiments are analyzing the model in qualitative and quantitative manner. First, as a qualitative analysis, the encoder filters are plotted varying the level of destruction, which shows that distinctive features were able to captured by adopting the concept of destruction. Second, as a quantitative analysis, the stacked denoising autoencoder with 3 hidden layers is compared to other models in classification task with several datasets. In most of the cases, the smallest error rates are achieved via the proposed model which verify its validity as a fine-tuning tool when training deep network.

2. Discussion

Here I would like to offer 2 discussion points. To begin with, rather than comparing the performance in classification, the quantitative analysis would have been also possible by measuring the disentanglement in terms of information theory such as Maximum Mean Discrepancy. Next, what if this is used along dropout and batch normalization? In my opinion, too much regularization a.k.a. stochasticity would require elaborate hyperparameter settings so that meta-learning should be utilized together.