Face Model Compression by Distilling Knowledge from Neurons

Ping Luo*, **Zhenyao Zhu***, **Ziwei Liu, Xiaogang Wang,** and **Xiaoou Tang**
Multimedia Lab, The Chinese University of Hong Kong
{pluo,zz012,lz013,xtang}@ie.cuhk.edu.hk, {xgwang}@ee.cuhk.edu.hk

Mean Field (MF) Method for Neuron Selection

Here we present the detailed derivations of the mean field (MF) method for neuron selection.

To start with, we consider the joint distribution $P(\mathbf{y})$ of MRF with energy $E(\mathbf{y})$:

$$E(\mathbf{y}) = \sum_{\forall i \in \mathcal{V}} \Phi(y_i) + \sum_{\forall i,j \in \mathcal{E}} \Psi(y_i, y_j) \tag{1}$$

$$P(\mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{y})\} \tag{2}$$

where $\Phi(y_i)$ is the unary term, $\Psi(y_i, y_j)$ is the pairwise term and Z is a partition function. MF method utilizes a fully-factorized proposal distribution $Q(\mathbf{y})$ to best approximate $P(\mathbf{y})$:

$$Q(\mathbf{y}) = \prod_{\forall i \in \mathcal{V}} p(y_i) \tag{3}$$

where $p(y_i)$ indicates the probability of choosing neuron i. The KL divergence between them is then calculated:

$$\begin{aligned} D_{KL}(Q\|P) &= \sum_{\mathbf{y}} Q(\mathbf{y}) \ln \left(\frac{Q(\mathbf{y})}{P(\mathbf{y})} \right) \\ &= \sum_{\mathbf{y}} Q(\mathbf{y}) E(\mathbf{y}) + \sum_{\mathbf{y}} Q(\mathbf{y}) \ln Q(\mathbf{y}) + \ln Z \end{aligned} \tag{4}$$

Since $\ln Z$ is a constant, minimizing the KL divergence between $Q(\mathbf{y})$ and $P(\mathbf{y})$ is equivalent to minimizing the former terms in Eqn.(4), which is denoted as free energy [2] $F(Q)$. And we can further substitute Eqn.(1) into it:

$$\begin{aligned} F(Q) &= \sum_{\mathbf{y}} Q(\mathbf{y}) E(\mathbf{y}) + \sum_{\mathbf{y}} Q(\mathbf{y}) \ln Q(\mathbf{y}) \\ &= \sum_{\forall i \in \mathcal{V}} p(y_i) \Phi(y_i) + \sum_{\forall i,j \in \mathcal{E}} p(y_i) p(y_j) \Psi_{ij} + \sum_{\forall i \in \mathcal{V}} p(y_i) \ln p(y_i) \end{aligned} \tag{5}$$

The final closed-form solution can be obtained by differentiating $F(Q)$ *w.r.t.* $p(y_i)$ and equating the resulting expression to zero:

$$p(y_i) \propto \exp\{-\big(\Phi(y_i) + \sum_{j=1, j \neq i}^{N} \sum_{y_j} p(y_j) \Psi(y_i, y_j)\big)\} \tag{6}$$

where N is the number of neurons to be selected. For efficiency, this iterative updating is performed in a coarse-to-fine manner.

Efficient Implementation of Mean Field Update

For a fully-connected graph with N nodes (a single model) or TN' nodes (an ensemble), the time complexities of the mean-field updates for them are $\mathcal{O}(N^2)$ and $\mathcal{O}((TN')^2)$ respectively, as mentioned in line 154-157 of the paper.

However, this procedure can be accelerated by using the piecewise-linear approximation [1]. Specifically, each element in $\mathbf{x}$ is first normalized to $[0, 1]$ and discretized into a set of values, *e.g.* {0.025,0.05,0.075,...}, whose distance matrix is then calculated and stored as an indexed table. In this case, the distance between any two continuous values is approximated by a linear interpolation between the distances of two closest discrete values, by looking up the table in $\mathcal{O}(1)$. As a result, Eqn.(6) can be efficiently solved in $\mathcal{O}(N)$ and $\mathcal{O}(TN')$ for a single model and an ensemble, respectively.

Attribute Classification Accuracy

For $\forall \mathbf{x}_i \in \mathbb{R}^{1 \times 40}$, we calculate the mean classification accuracy of the j-th attribute $\mathbf{x}_{i(j)}$ within two steps. In the first step, we collect response for each image in the validation set with respect to neuron i, denoted as $r_i(\mathbf{I})$, which can be viewed as a decision score for attribute classification. In the second step, the mean accuracy $\mathbf{x}_{i(j)}$ is computed as the mean of the true positive and true negative rates:

$$\mathbf{x}_{i(j)} = \frac{TP + TN}{2} \tag{7}$$

$$TP = \frac{\sum_{k=1}^{N^+} \mathbf{1}(r_i(\mathbf{I}_k) \geq \epsilon_i^j)}{N^+} \tag{8}$$

$$TN = \frac{\sum_{k=1}^{N^-} \mathbf{1}(r_i(\mathbf{I}_k) < \epsilon_i^j)}{N^-} \tag{9}$$

where N^+ and N^- represent the positive and negative samples respectively, while ϵ_i^j is a decision threshold determined by greedily searching over the entire validation set. In contrast to raw accuracy, the mean accuracy as above is able to reduce dataset biases between positive and negative samples.

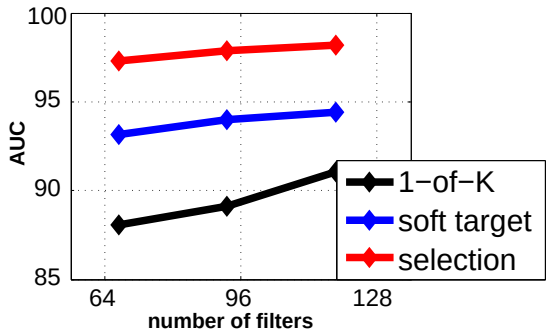

Figure 1: Comparisons of performances with respect to different number of filters.

More Results

Fig.1 shows that when the width (number of filters in the convolutional layers) of a student decreases, its performance also decreases. We employ three students as representatives, including 'S-1-of-K', 'S-soft target (t=1)', and 'S-selection'. The decreasing tendencies of the students supervised by soft target and neuron selection are relative small, compared to that of the 1-of-K label. More informative labels are easier to train. Neuron selection achieves the best result with respect to different number of filters. The performance gap between them enlarges when the number of filters decreases.

References

[1] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH*, 2002.

[2] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.