Distilling the Knowledge in a Neural Network

Geoffrey Hinton, Oriol Vinyals, Jeff Dean

March 2015

Content

- Distillation
- Matching logits is a special case of distillation
- Preliminary experiments on MNIST
- Experiments on speech recognition
- Soft Targets as Regularizers

Distillation

- Neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit, computed for each class into a probability, by comparing it with the other logits

  $$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)},$$

  where T is a temperature that is normally set to 1. Using a higher value for T produces a softer probability distribution over classes.
- In the simplest form of distillation, knowledge is transferred to the distilled model by training it on a transfer set and using a soft target distribution for each case in the transfer set that is produced by using the cumbersome model with a high temperature in its softmax. The same high temperature is used when training the distilled model, but after it has been trained it uses a temperature of 1.

Distilling

- When the correct labels are known for all or some of the transfer set, this method can be significantly improved by also training the distilled model to produce the correct labels. The better way is to simply use a weighted average of two different objective functions. The first objective function is the cross entropy with the soft targets and this cross entropy is computed using the same high temperature in the softmax of the distilled model as was used for generating the soft targets from the cumbersome model. The second objective function is the cross entropy with the correct labels. This is computed using exactly the same logits in softmax of the distilled model but at a temperature of 1. Authors found that the best results were generally obtained by using a considerably lower weight on the second objective function.

Matching logits is a special case of distillation

- Each case in the transfer set contributes a cross-entropy gradient, dC/dz_i , with respect to each logit, z_i of the distilled model. If the cumbersome model has logits v_i which produce soft target probabilities p_i and the transfer training is done at a temperature of T , this gradient is given by:

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)$$

- If the temperature is high compared with the magnitude of the logits, we can approximate:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right)$$

Matching logits is a special case of distillation

- If we now assume that the logits have been zero-meaned separately for each transfer case so that $\sum_j z_j = \sum_j v_j = 0$ equation simplifies to

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i)$$

- So in the high temperature limit, distillation is equivalent to minimizing $1/2\,(z_i - v_i)^2$, provided the logits are zero-meaned separately for each transfer case. At lower temperatures, distillation pays much less attention to matching logits that are much more negative than the average. This is potentially advantageous because these logits are almost completely unconstrained by the cost function used for training the cumbersome model so they could be very noisy.

Preliminary experiments on MNIST

- Authors trained a single large neural net with two hidden layers of 1200 rectified linear hidden units on all 60,000 training cases. The net was strongly regularized using dropout and weight-constraints. This net achieved 67 test errors whereas a smaller net with two hidden layers of 800 rectified linear hidden units and no regularization achieved 146 errors. But if the smaller net was regularized solely by adding the additional task of matching the soft targets produced by the large net at a temperature of 20, it achieved 74 test errors.

Experiments on speech recognition

- Authors used an architecture with 8 hidden layers each containing 2560 rectified linear units and a final softmax layer with 14,000 labels (HMM targets ht). The input was 26 frames of 40 Mel-scaled filterbank coefficients with a 10ms advance per frame and they predicted the HMM state of 21st frame. The total number of parameters was about 85M. This was a slightly outdated version of the acoustic model used by Android voice search, and should be considered as a very strong baseline. To train the DNN acoustic model they used about 2000 hours of spoken English data, which yields about 700M training examples.

System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled single model	60.8%	10.7%

Soft Targets as Regularizers

- One of main claims about using soft targets instead of hard targets is that a lot of helpful information can be carried in soft targets that could not possibly be encoded with a single hard target. In this section authors demonstrated that this is a very large effect by using far less data to fit the 85M parameters of the baseline speech model described earlier.

System & training set	Train Frame Accuracy	Test Fra
Baseline (100% of training set)	63.4%	5
Baseline (3% of training set)	67.3%	4
Soft Targets (3% of training set)	65.4%	5

The End