Appendix: ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning

Quoc V. Le, Alexandre Karpenko, Jiquan Ngiam and Andrew Y. Ng
$\{$*quocle,akarpenko,jngiam,ang*$\}$*@cs.stanford.edu*
Computer Science Department, Stanford University

1 Proofs and Discussions of Lemmas

Proof of Lemma 3.1: We have:

$$\frac{1}{m}\sum_{i=1}^{m}\|W^TWx^{(i)}-x^{(i)}\|_2^2 = \text{tr}\Big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})\frac{1}{m}\sum_{i=1}^{m}x^{(i)}(x^{(i)})^T\Big] = \|W^TW-\mathbf{I}\|_{\mathcal{F}}^2$$

Note, since the data is whitened: $\mathbb{E}\big[xx^T\big] = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}(x^{(i)})^T = \mathbf{I}$. $\blacksquare$

From the proof, it can also be seen that when the data is not whitened, $\frac{1}{m}\sum_{i=1}^{m}\|W^TWx^{(i)} - x^{(i)}\|_2^2 = \text{tr}\big[(W^TW - \mathbf{I})^TC_x(W^TW - \mathbf{I})\big]$. Further, by diagonalizing C_x, we have $\frac{1}{m}\sum_{i=1}^{m}\|W^TWx^{(i)}-x^{(i)}\|_2^2 = \text{tr}\big[(W^TW-\mathbf{I})EDE^T(W^TW-\mathbf{I})^T\big] = \big\|\big[(W^TW-\mathbf{I})ED^{\frac{1}{2}}\big\|_{\mathcal{F}}^2$. Here, C_x denotes the covariance matrix of the data. E is the matrix whose columns are eigenvectors of C_x; and D is a diagonal matrix of eigenvalues of C_x.

This result also means that when the data is not whitened, the reconstruction cost becomes a weighted linear regression in the space rotated by eigenvectors and scaled by eigenvalues. The cost is therefore weighted heavily in the principal vector directions and less so in the other directions. Also, note that $D^{-\frac{1}{2}}E^T$ is the well-known PCA whitening matrix. The optimization therefore builds in an inverse whitening matrix such that W will have to learn the whitening matrix to cancel out $ED^{\frac{1}{2}}$.

Proof of Lemma 3.2: We have

$$\|WW^T - \mathbf{I}_k\|_{\mathcal{F}}^2 = \text{tr}\Big[(WW^T - \mathbf{I}_k)^T(WW^T - \mathbf{I}_k)\Big] \tag{1}$$

$$= \text{tr}(WW^TWW^T) - 2\text{tr}(WW^T) + \text{tr}(\mathbf{I}_k) = \text{tr}(W^TWW^TW) - 2\text{tr}(W^TW) + \text{tr}(\mathbf{I}_n) + k - n \tag{2}$$

$$= \text{tr}\Big[W^TWW^TW - 2W^TW + \mathbf{I}_n\Big] + k - n = \text{tr}\Big[(W^TW - \mathbf{I}_n)^T(W^TW - \mathbf{I}_n)\Big] + k - n \tag{3}$$

$$= \|W^TW - \mathbf{I}_n\|_{\mathcal{F}}^2 + k - n \tag{4}$$

Note, in step (3), we used the fact that tr$(I_k) = k$ and tr$(I_n) = n$. In the second part of step (3), we used the property tr$(AB) =$ tr(BA) if AB is square. $\blacksquare$

Finally, we present a third lemma (a generalization of Lemma 3.1) that shows the equivalence of denoising autoencoders and RICA with weight decay for whitened data. Denoising autoencoders are often preferred over standard autoencoders. In denoising autoencoders, the algorithm is asked to reconstruct the original input given its corrupted version.

Lemma 3.3 *The denoising reconstruction cost with additive noise* $\frac{1}{m}\sum_{i=1}^{m}\|W^TW(x^{(i)} + \epsilon^{(i)}) - x^{(i)}\|_2^2$*, where* $\epsilon^{(i)}$ *are random variables with zero mean and variance* $\delta^2\mathbf{I}$*, is equivalent to* $\|(W^TW - \mathbf{I})E(D + \delta^2\mathbf{I})^{\frac{1}{2}}\|_{\mathcal{F}}^2 + 2\delta^2\sum_{j=1}^{k}\|W_j\|_2^2$ *up to an additive constant.*

If the data is whitened, then E and D are both identity matrices. As a result, denoising is equivalent to simply adding a weight decay penalty to objective functions of RICA.

Proof of Lemma 3.3: We have:

$$\frac{1}{m}\sum_{i=1}^{m}\|W^TW(x^{(i)}+\epsilon^{(i)})-x^{(i)}\|_2^2$$

$$=\frac{1}{m}\sum_{i=1}^{m}\|(W^TW-\mathbf{I})x^{(i)}+W^TW\epsilon^{(i)}\|_2^2 \tag{5}$$

$$\begin{aligned}&=\frac{1}{m}\sum_{i=1}^{m}\text{tr}\big[(x^{(i)})^T(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})x^{(i)}\big]\\&\quad+2\text{tr}\big[(\epsilon^{(i)})^TW^TW(W^TW-\mathbf{I})x^{(i)}\big]+\text{tr}\big[(\epsilon^{(i)})^TW^TWW^TW\epsilon^{(i)})\big]\end{aligned} \tag{6}$$

$$\begin{aligned}&=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})\frac{1}{m}\sum_{i=1}^{m}x^{(i)}(x^{(i)})^T\big]\\&\quad+2\text{tr}\big[W^TW(W^TW-\mathbf{I})\frac{1}{m}\sum_{i=1}^{m}x^{(i)}(\epsilon^{(i)})^T\big]+\text{tr}\big[W^TWW^TW\frac{1}{m}\sum_{i=1}^{m}\epsilon^{(i)}(\epsilon^{(i)})^T)\big]\end{aligned} \tag{7}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})C_x\big]+\text{tr}\big[W^TWW^TW\delta^2\mathbf{I}\big] \tag{8}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})C_x\big]+\text{tr}\big[((W^TW-\mathbf{I})^T(W^TW-\mathbf{I})+2W^TW-\mathbf{I})\delta^2\mathbf{I}\big] \tag{9}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})C_x\big]+\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})\delta^2\mathbf{I}\big]+\delta^2\text{tr}\big[(2W^TW-\mathbf{I})\big] \tag{10}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})(C_x+\delta^2\mathbf{I})\big]+\delta^2\text{tr}\big[(2W^TW)\big]+c \tag{11}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})(C_x+\delta^2\mathbf{I})\big]+2\delta^2\text{tr}\big[(W^TW)\big]+c \tag{12}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})(EDE^T+E\delta^2\mathbf{I}E^T)\big]+2\delta^2\text{tr}\big[(W^TW)\big]+c \tag{13}$$

$$=\text{tr}\big[(W^TW-\mathbf{I})^T(W^TW-\mathbf{I})E(D+\delta^2\mathbf{I})E^T\big]+2\delta^2\text{tr}\big[(W^TW)\big]+c \tag{14}$$

$$=\|(W^TW-\mathbf{I})E(D+\delta^2\mathbf{I})^{\frac{1}{2}}\|_{\mathcal{F}}^2+2\delta^2\sum_{j=1}^{k}\|W_j\|_2^2+c \tag{15}$$

We note that x and ϵ are uncorrelated: $\mathbb{E}\big[x\epsilon^T\big]=\frac{1}{m}\sum_{i=1}^{m}x^{(i)}(\epsilon^{(i)})^T=\mathbf{0}$ and c is a constant. ∎

2 Norm ball projection

When the representations are overcomplete $k >> n$, notice that it is possible to reconstruct the data ($\sum_{i=1}^{m}\|W^TWx_i-x_i\|_2^2$), with only a complete subset of the rows of W, while setting the rest to zero. Therefore, in order to learn an overcomplete representation without degenerate (zero) features, we constrain each row of W to have a L_2 norm of 1, i.e.,

$$\|W_i\|_2^2=1, \forall i=1,\ldots,k. \tag{16}$$

Although these constraints prevent learning degenerate representations, they make the optimization problem potentially harder and require the use of projected gradient methods.

In order to use L-BFGS/CG to solve this constrained optimization problem we employ L_2-norm ball projection. The key idea is project the weights onto the norm ball during each iteration of the optimization. Specifically, we let $\hat{W}_i=W_i/\|W_i\|_2^2$.

Since optimization methods such as L-BFGS or CG evaluate the objective function using the supplied gradient vector, we need to account for the projection in the gradient computation. In particular, we will need to "backpropagate" the gradients through the projection, thereby taking into account the projection during the optimization. This allows us to treat the optimization problem as if it is unconstrained, which makes optimization run much faster.[1]

[1] While W could potentially grow unbounded, we do not find that this is a problem in practice, because simple scaling of W will not be a descent direction. Furthermore, one could always rescale W to be on the norm ball in the outer loop of the optimization method.

The gradient computation that takes projection into account is carried out as follows:

Procedure 1 RICA for large overcomplete representations

```
  Step 1 (Projection): $\hat{W}_{ij} = \frac{W_{ij}}{\sqrt{\epsilon + \sum_{l=1}^{n} W_{il}^2}}, \; \forall i = 1, \ldots, k$.
  Step 2: Compute gradient $g$ using $\hat{W}$.
  Step 3 (Inverse Projection): $g_{ij} = \frac{g_{ij}}{\sqrt{\epsilon + \sum_{l=1}^{n} W_{il}^2}} - \hat{W}_{ij} \frac{\sum_{l=1}^{n} g_{il} W_{il}}{\epsilon + \sum_{l=1}^{n} W_{il}^2}$
```