Autoencoder Tuning with PCA Principles

According to (1), “Autoencoders are directly related to Principal Component Analysis (PCA),” and an Autoencoder “extends PCA to a nonlinear space.” Mathematically correct autoencoders are “well-posed,” meaning they inherit the following properties of PCA:

- Tied Weights
  - Weights are equal for encoder and corresponding decoder layer (1)
  - Decoder weights are the transpose of the encoder weights, and this reduces the number of parameters for the model (2)
- Orthogonal Weights
  - Each weight vector is independent of others (1)
- Uncorrelated Features
  - Output of encoding layer is not correlated (1)
- Unit Norm
  - Weights on a layer have unit norm (1)

Furthermore, (1) explains how incorporating these properties provides regularization, addresses “exploding and vanishing gradient,” and creates a smaller network for edge computing.

I am still exploring how to properly implement orthogonal weights and uncorrelated features.

Unit Norm

```
Encoder weights norm
[0.995 0.991 1.009 1.011 0.992 1.005 1.    1.001 1.006 1.001]

[1.004 1.006 0.996 1.003 1.006 0.997]

Decoder weights norm
[1.004 1.006 0.996 1.003 1.006 0.997]

[0.995 0.991 1.009 1.011 0.992 1.005 1.    1.001 1.006 1.001]
```

Tied Weights

```
===Encoder weights===
Hidden layer
[[ 0.223 -0.353 -0.272  0.392  0.133  0.229 -0.168 -0.244  0.449  0.01
  -0.488]
 [-0.362  0.22   0.25  -0.384 -0.055  0.184 -0.358 -0.195  0.2   -0.222
   0.564]
 [ 0.262  0.206 -0.237 -0.333  0.231  0.367  0.317  0.336  0.158  0.135
  -0.527]
 [ 0.166 -0.029  0.496  0.379  0.132  0.451  0.171  0.416  0.127  0.086
   0.368]
 [ 0.925  0.265  0.01   0.253  0.045 -0.007  0.043 -0.013  0.027 -0.064
  -0.038]
 [ 0.235  0.168 -0.274  0.102 -0.012 -0.136  0.042 -0.005  0.668  0.074
   0.6  ]
 [-0.592  0.     0.487 -0.14   0.436 -0.082  0.313  0.213  0.015 -0.107
   0.201]
 [-0.546 -0.463 -0.469 -0.183 -0.153  0.03  -0.302  0.055 -0.164 -0.288
   0.076]
 [ 0.369 -0.14  -0.293 -0.086  0.087  0.221  0.154 -0.035 -0.726  0.123
   0.358]
 [ 0.25   0.413 -0.326  0.301 -0.339 -0.017 -0.417 -0.01  -0.082  0.314
   0.42 ]]

Latent layer
[[ 0.102  0.005  0.301  0.268  0.499  0.064  0.022 -0.538 -0.162 -0.507]
 [ 0.156  0.02   0.385  0.188 -0.339 -0.468  0.207  0.123 -0.514 -0.367]
 [-0.292 -0.196  0.134 -0.367 -0.183 -0.669 -0.153  0.371  0.105  0.265]
 [ 0.055  0.167  0.236  0.45  -0.329  0.54   0.032  0.201  0.224 -0.469]
 [ 0.202  0.46  -0.278 -0.24  -0.012  0.204 -0.397  0.02   0.365  0.529]
 [-0.264  0.201 -0.328  0.117 -0.187 -0.01   0.43  -0.093  0.621 -0.393]]
```

```
===Decoder weights===
Hidden layer 2
[[ 0.102  0.005  0.301  0.268  0.499  0.064  0.022 -0.538 -0.162 -0.507]
 [ 0.156  0.02   0.385  0.188 -0.339 -0.468  0.207  0.123 -0.514 -0.367]
 [-0.292 -0.196  0.134 -0.367 -0.183 -0.669 -0.153  0.371  0.105  0.265]
 [ 0.055  0.167  0.236  0.45  -0.329  0.54   0.032  0.201  0.224 -0.469]
 [ 0.202  0.46  -0.278 -0.24  -0.012  0.204 -0.397  0.02   0.365  0.529]
 [-0.264  0.201 -0.328  0.117 -0.187 -0.01   0.43  -0.093  0.621 -0.393]]

Output layer
[[ 0.223 -0.353 -0.272  0.392  0.133  0.229 -0.168 -0.244  0.449  0.01
  -0.488]
 [-0.362  0.22   0.25  -0.384 -0.055  0.184 -0.358 -0.195  0.2   -0.222
   0.564]
 [ 0.262  0.206 -0.237 -0.333  0.231  0.367  0.317  0.336  0.158  0.135
  -0.527]
 [ 0.166 -0.029  0.496  0.379  0.132  0.451  0.171  0.416  0.127  0.086
   0.368]
 [ 0.925  0.265  0.01   0.253  0.045 -0.007  0.043 -0.013  0.027 -0.064
  -0.038]
 [ 0.235  0.168 -0.274  0.102 -0.012 -0.136  0.042 -0.005  0.668  0.074
   0.6  ]
 [-0.592  0.     0.487 -0.14   0.436 -0.082  0.313  0.213  0.015 -0.107
   0.201]
 [-0.546 -0.463 -0.469 -0.183 -0.153  0.03  -0.302  0.055 -0.164 -0.288
   0.076]
 [ 0.369 -0.14  -0.293 -0.086  0.087  0.221  0.154 -0.035 -0.726  0.123
   0.358]
 [ 0.25   0.413 -0.326  0.301 -0.339 -0.017 -0.417 -0.01  -0.082  0.314
   0.42 ]]
```

Near-Orthogonality (from Unit Norm constraint)

```
Encoder weights dot products
[[ 1.    -0.42  0.23 -0.07  0.24  0.08 -0.48  0.01 -0.28 -0.1 ]
 [-0.42  1.    -0.4   0.06 -0.39  0.26  0.34  0.23 -0.19  0.12]
 [ 0.23 -0.4   1.    0.02  0.24 -0.12 -0.1  -0.27  0.02 -0.28]
 [-0.07  0.06  0.02  1.    0.24  0.19  0.32 -0.43  0.06  0.03]
 [ 0.24 -0.39  0.24  0.24  1.    0.28 -0.55 -0.69  0.25  0.34]
 [ 0.08  0.26 -0.12  0.19  0.28  1.    -0.15 -0.2  -0.15  0.46]
 [-0.48  0.34 -0.1   0.32 -0.55 -0.15  1.    0.01 -0.24 -0.58]
 [ 0.01  0.23 -0.27 -0.43 -0.69 -0.2   0.01  1.    0.07 -0.1 ]
 [-0.28 -0.19  0.02  0.06  0.25 -0.15 -0.24  0.07  1.    0.25]
 [-0.1   0.12 -0.28  0.03  0.34  0.46 -0.58 -0.1   0.25  1.  ]]

[[ 1.    0.19 -0.58  0.16 -0.47 -0.03]
 [ 0.19  1.    0.17  0.13 -0.67 -0.17]
 [-0.58  0.17  1.   -0.51  0.01 -0.15]
 [ 0.16  0.13 -0.51  1.   -0.15  0.37]
 [-0.47 -0.67  0.01 -0.15  1.   -0.05]
 [-0.03 -0.17 -0.15  0.37 -0.05  1.  ]]

Decoder weights dot product
[[ 1.    0.19 -0.58  0.16 -0.47 -0.03]
 [ 0.19  1.    0.17  0.13 -0.67 -0.17]
 [-0.58  0.17  1.   -0.51  0.01 -0.15]
 [ 0.16  0.13 -0.51  1.   -0.15  0.37]
 [-0.47 -0.67  0.01 -0.15  1.   -0.05]
 [-0.03 -0.17 -0.15  0.37 -0.05  1.  ]]

[[ 1.    -0.42  0.23 -0.07  0.24  0.08 -0.48  0.01 -0.28 -0.1 ]
 [-0.42  1.    -0.4   0.06 -0.39  0.26  0.34  0.23 -0.19  0.12]
 [ 0.23 -0.4   1.    0.02  0.24 -0.12 -0.1  -0.27  0.02 -0.28]
 [-0.07  0.06  0.02  1.    0.24  0.19  0.32 -0.43  0.06  0.03]
 [ 0.24 -0.39  0.24  0.24  1.    0.28 -0.55 -0.69  0.25  0.34]
 [ 0.08  0.26 -0.12  0.19  0.28  1.    -0.15 -0.2  -0.15  0.46]
 [-0.48  0.34 -0.1   0.32 -0.55 -0.15  1.    0.01 -0.24 -0.58]
 [ 0.01  0.23 -0.27 -0.43 -0.69 -0.2   0.01  1.    0.07 -0.1 ]
 [-0.28 -0.19  0.02  0.06  0.25 -0.15 -0.24  0.07  1.    0.25]
 [-0.1   0.12 -0.28  0.03  0.34  0.46 -0.58 -0.1   0.25  1.  ]]
```

Sources

1. https://towardsdatascience.com/build-the-right-autoencoder-tune-and-optimize-using-pca-principles-part-ii-24b9cca69bd6
2. https://medium.com/@lmayrandprovencher/building-an-autoencoder-with-tied-weights-in-keras-c4a559c529a2