Supplementary Material for Paper "Joint autoencoders: a flexible multi-task learning framework"

Implementation details

All the images are scaled to $[0, 1]$. In all cases, the training is done using the ADAM optimizer with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The Keras default Xavier initialization is used. Shared layers are denoted in red and connected by a bidirectional arrow: $\leftrightarrow$. $Conv\, n \times (k \times k)$ stands for a convolution layer with n filters of size $k \times k$. $ReLU$ stands for a rectified linear unit, i.e. the function $\max(x, 0)$. $MP\, k \times k\, stride\, s$ stands for max-pooling of size $k \times k$ with stride s. $FC\, k$ stands for a fully-connected layer of size k. The symbol $\oplus$ stands for the merge operation. For instance, if it appears after fully-connected layers of size 500 each, it denotes the resulting merged layer of size 1000. Outputs are processed by a SoftMax.

Unsupervised learning - MNIST

For the MNIST reconstruction experiments, we utilize a CNN-based version of the autoencoder and JAE presented in Figure 1 in the body. Mini-batch size is set to 256, with 10 epochs. The JAE losses are weighed equally.

Figure 1: An MNIST autoencoder. A pair of these is used as a benchmark for the MNIST joint autoencoder.

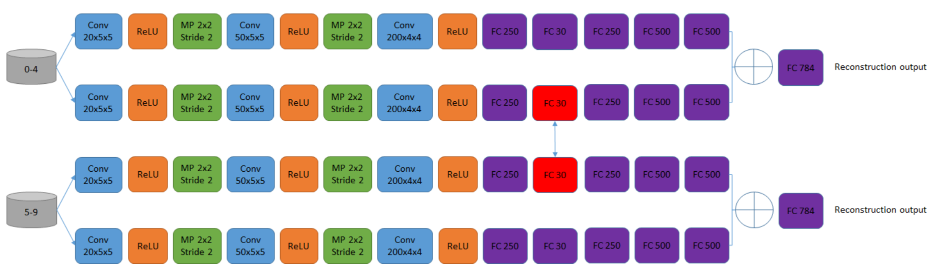

Figure 2: A joint autoencoder for reconstruction of MNIST subsets

Unsupervised learning - CIFAR-10

Mini-batch size is set to 128, with 10 epochs. The JAE losses are weighed equally. "Deconv $n \times k \times k$" stands for a deconvolution layer with n filters of size $k \times k$ with 2×2 upsampling.

Figure 3: A CIFAR-10 autoencoder. A pair of these is used as a benchmark for the CIFAR-10 joint autoencoder.

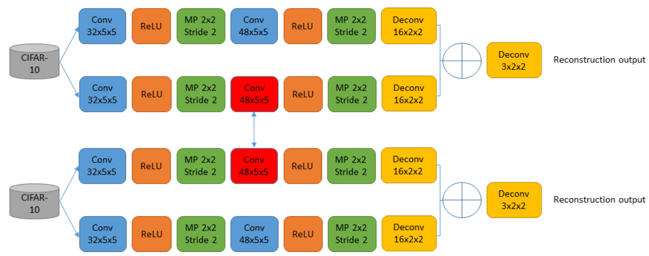

Figure 4: A joint autoencoder for reconstruction of CIFAR-10 subsets

Unsupervised learning - celebA

The images are rescaled to $64 \times 64 \times 3$. Mini-batch size is set to 64, with 30 epochs. The JAE losses are weighed equally. We omit the ReLU activations for brevity, and use "Conv n, k, s" and "Deconv n, k, s" to denote convolutions and deconvolutions with n filters of size k with strides s.

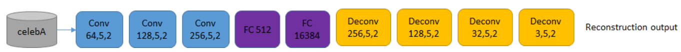

Figure 5: A celebA autoencoder. A pair of these is used as a benchmark for the celebA joint autoencoder.

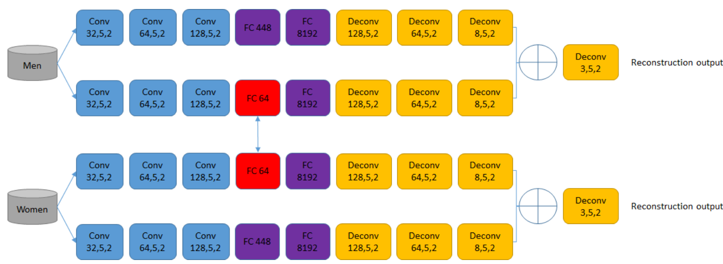

Figure 6: A joint autoencoder for reconstruction of celebA subsets

Transfer learning - MNIST↔USPS

Mini-batch size is set to 64, with 10 epochs. The reconstruction losses are weighed 4 times higher than the classification losses.

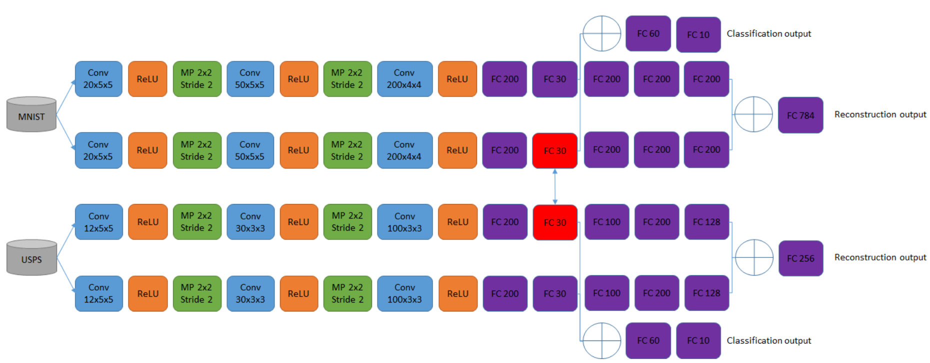

Figure 7: An MNIST-USPS joint autoencoder

Transfer learning - SVHN→MNIST

Mini-batch size is set to 64, with 10 epochs. The reconstruction losses are weighed 4 times lower than the classification losses. In this case, as opposed to the previous one, the classification task is challenging enough to avoid early overfitting.

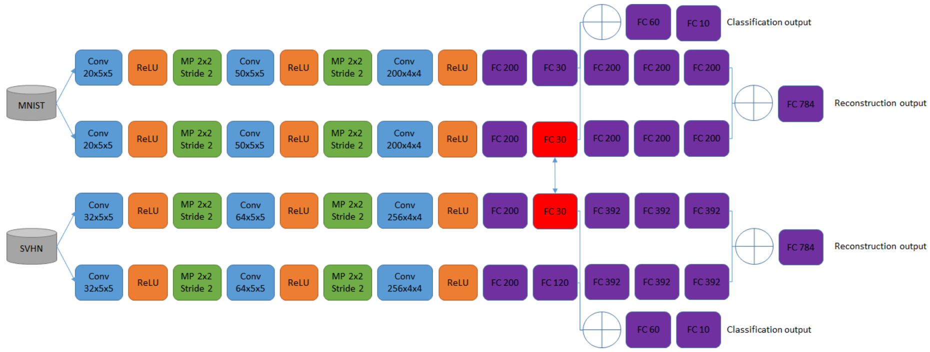

Figure 8: A joint autoencoder for transfer learning from SVHN to MNIST

Transfer learning - SVHN→MNIST+USPS

Mini-batch size is set to 64, with 20 epochs. The reconstruction losses are weighed 4 times lower than the classification losses.

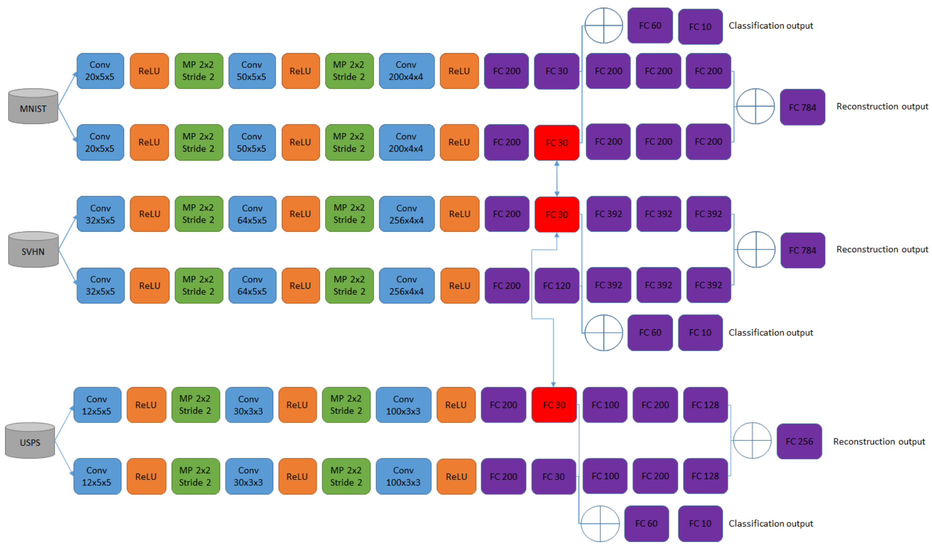

Figure 9: A three-way joint autoencoder for transfer learning from SVHN to MNIST and to USPS