Supplementary material for the submission #1536

1. An alternative optimization approach

There exists an alternative construction (inspired by (Goodfellow et al., 2014)) that leads to the same updates (4)-(6). Rather than using the gradient reversal layer, the construction introduces two different loss functions for the domain classifier. Minimization of the first domain loss (L_{d+}) should lead to a better domain discrimination, while the second domain loss (L_{d-}) is minimized when the domains are distinct. Stochastic updates for θ_f and θ_d are then defined as:

$$\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial L_y^i}{\partial\theta_f} + \frac{\partial L_{d-}^i}{\partial\theta_f}\right)$$

$$\theta_d \longleftarrow \theta_d - \mu\frac{\partial L_{d+}^i}{\partial\theta_d},$$

Thus, different parameters participate in the optimization of different losses

In this framework, the gradient reversal layer constitutes a special case, corresponding to the pair of domain losses $(L_d, -\lambda L_d)$. However, other pairs of loss functions can be used. One example would be the binomial cross-entropy (Goodfellow et al., 2014):

$$L_{d+}(q, d) = \sum_{i=1..N} d_i \log(q_i) + (1 - d_i)\log(1 - q_i),$$

where d indicates domain indices and q is an output of the predictor. In that case "adversarial" loss is easily obtained by swapping domain labels, i.e. $L_{d-}(q, d) = L_{d+}(q, 1-d)$. This particular pair has a potential advantage of producing stronger gradients at early learning stages if the domains are quite dissimilar. In our experiments, however, we did not observe any significant improvement resulting from this choice of losses.

2. CNN architectures

Four different architectures were used in our experiments (first three are shown in Figure 1):

- A smaller one (a) if the source domain is MNIST. This architecture was inspired by the classical LeNet-5 (LeCun et al., 1998).
- (b) for the experiments involving SVHN dataset. This one is adopted from (Srivastava et al., 2014).
- (c) in the SYN SIGNS $\rightarrow$ GTSRB setting. We used the single-CNN baseline from (Cireşan et al., 2012) as our starting point.
- Finally, we use pre-trained `AlexNet` from the `Caffe`-package (Jia et al., 2014) for the OFFICE domains. Adaptation architecture is identical to (Tzeng et al., 2014): 2-layer domain classifier ($x \rightarrow 1024 \rightarrow 1024 \rightarrow 2$) is attached to the 256-dimensional bottleneck of `fc7`.

The domain classifier branch in all cases is somewhat arbitrary (better adaptation performance might be attained if this part of the architecture is tuned).

3. Training procedure

We use stochastic gradient descent with 0.9 momentum and the learning rate annealing described by the following formula:

$$\mu_p = \frac{\mu_0}{(1 + \alpha \cdot p)^\beta},$$

where p is the training progress linearly changing from 0 to 1, $\mu_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$ (the schedule was optimized to promote convergence and low error on the *source* domain).

Following (Srivastava et al., 2014) we also use dropout and ℓ_2-norm restriction when we train the SVHN architecture.

References

Cireşan, Dan, Meier, Ueli, Masci, Jonathan, and Schmidhuber, Jürgen. Multi-column deep neural network for traffic sign classification. *Neural Networks*, (32):333–338, 2012.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *NIPS*, 2014.

Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, Jonathan, Girshick, Ross, Guadarrama, Sergio, and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout:

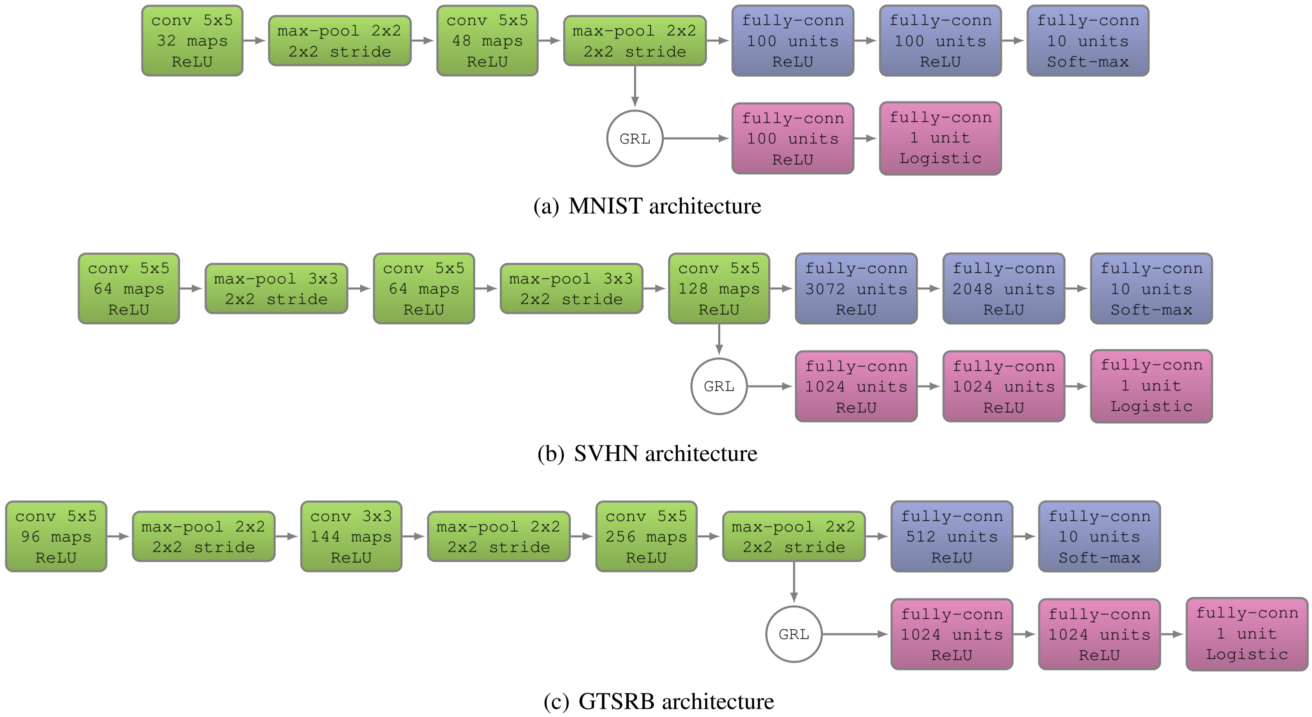

Figure 1. CNN architectures used in the experiments. Boxes correspond to transformations applied to the data. Color-coding is the same as in Figure 1.

A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Tzeng, Eric, Hoffman, Judy, Zhang, Ning, Saenko, Kate, and Darrell, Trevor. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.