Neural networks
Deep learning - example
Topics: pseudocode

• for \( l = 1 \) to \( L \)
  
  ‣ build unsupervised training set (with \( h^{(0)}(x) = x \ )):

  \[
  \mathcal{D} = \left\{ h^{(l-1)}(x^{(t)}) \right\}_{t=1}^{T}
  \]

  ‣ train “greedy module” (RBM, autoencoder) on \( \mathcal{D} \)

  ‣ use hidden layer weights and biases of greedy module to initialize the deep network parameters \( W^{(l)}, b^{(l)} \)

• Initialize \( W^{(L+1)}, b^{(L+1)} \) randomly (as usual)

• Train the whole neural network using (supervised) stochastic gradient descent (with backprop)
Topics: pseudocode

- for \( l = 1 \) to \( L \)
  - build unsupervised training set (with \( h^{(0)}(x) = x \)):
    \[
    D = \left\{ h^{(l-1)}(x^{(t)}) \right\}_{t=1}^{T}
    \]
  - train “greedy module” (RBM, autoencoder) on \( D \)
  - use hidden layer weights and biases of greedy module to initialize the deep network parameters \( W^{(l)}, b^{(l)} \)

- Initialize \( W^{(L+1)}, b^{(L+1)} \) randomly (as usual)
- Train the whole neural network using (supervised) stochastic gradient descent (with backprop)
**Topics:** pseudocode

- for \( l = 1 \) to \( L \)
  - build unsupervised training set (with \( h^{(0)}(x) = x \)):
    \[
    \mathcal{D} = \left\{ h^{(l-1)}(x^{(t)}) \right\}^T_{t=1}
    \]
  - train “greedy module” (RBM, autoencoder) on \( \mathcal{D} \)
  - use hidden layer weights and biases of greedy module to initialize the deep network parameters \( W^{(l)}, b^{(l)} \)

- Initialize \( W^{(L+1)}, b^{(L+1)} \) randomly (as usual)
- Train the whole neural network using (supervised) stochastic gradient descent (with backprop)

\[ \text{pre-training} \]
\[ \text{fine-tuning} \]
**Topics:** datasets

- Datasets generated with varying number of factors of variations

Variations on MNIST

- MNIST-rotation
- MNIST-random-background
- MNIST-image-background
- MNIST-background-rotation

Tall or wide?

Convex shape or not?

An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation
Larochelle, Erhan, Courville, Bergstra and Bengio, 2007
## DEEP LEARNING

**Topics:** impact of initialization

<table>
<thead>
<tr>
<th>Network</th>
<th>Depth</th>
<th>MNIST-small classif. test error</th>
<th>MNIST-rotation classif. test error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deep net</strong></td>
<td>1</td>
<td>4.14% ± 0.17</td>
<td>15.22% ± 0.31</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4.03% ± 0.17</td>
<td>10.63% ± 0.27</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4.24% ± 0.18</td>
<td>11.98% ± 0.28</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4.47% ± 0.18</td>
<td>11.73% ± 0.29</td>
</tr>
<tr>
<td><strong>Deep net + autoencoder</strong></td>
<td>1</td>
<td>3.87% ± 0.17</td>
<td>11.43% ± 0.28</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.38% ± 0.16</td>
<td>9.88% ± 0.26</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.37% ± 0.16</td>
<td>9.22% ± 0.25</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.39% ± 0.16</td>
<td>9.20% ± 0.25</td>
</tr>
<tr>
<td><strong>Deep net + RBM</strong></td>
<td>1</td>
<td>3.17% ± 0.15</td>
<td>10.47% ± 0.27</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.74% ± 0.14</td>
<td>9.54% ± 0.26</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2.71% ± 0.14</td>
<td>8.80% ± 0.25</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2.72% ± 0.14</td>
<td>8.83% ± 0.24</td>
</tr>
</tbody>
</table>
Why Does Unsupervised Pre-training Help Deep Learning?
Erhan, Bengio, Courville, Manzagol, Vincent and Bengio, 2011
Topics: impact of initialization

Fig. 9: Effect of layer size on the changes brought by unsupervised pre-training, for networks with 1, 2 or 3 hidden layers. Experiments on MNIST. Error bars have a height of two standard deviations (over initialization seed). Pre-training hurts for smaller layer sizes and shallower networks, but it helps for all depths for larger networks.

The small size of the hidden layers. As the model size decreases from 800 hidden units, the generalization error increases, and it increases more with unsupervised pre-training presumably because of the extra regularization effect: small networks have a limited capacity already so further restricting it (or introducing an additional bias) can harm generalization. Such a result seems incompatible with a pure optimization effect. We also obtain the result that DBNs and SDAEs seem to have qualitatively similar effects as pre-training strategies.

The effect can be explained in terms of the role of unsupervised pre-training as promoting input transformations (in the hidden layers) that are useful at capturing the main variations in the input distribution $P(X)$. It may be that only a small subset of these variations are relevant for predicting the class label $Y$. When the hidden layers are small it is less likely that the transformations for predicting $Y$ are included in the lot learned by unsupervised pre-training.

7.4 Experiment 4: Challenging the Optimization Hypothesis

Experiments 1–3 results are consistent with the regularization hypothesis and Experiments 2–3 would appear to directly support the regularization hypothesis over the alternative—that unsupervised pre-training aids in optimizing the deep model objective function.

In the literature there is some support for the optimization hypothesis. Bengio et al. (2007) constrained the top layer of a deep network to have 20 units and measured the training error of networks with and without pre-training. The idea was to prevent the networks from overfitting the training error simply with the top hidden layer, thus to make it clearer whether some optimization helps.
Topics: choice of hidden layer size
# Deep Learning

**Topics:** performance on different datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM$_{rbf}$</th>
<th>SAA-3</th>
<th>DBN-3</th>
<th>SdA-3 (ν)</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td><strong>3.03±0.15</strong></td>
<td>3.46±0.16</td>
<td>3.11±0.15</td>
<td><strong>2.80±0.14</strong> (10%)</td>
</tr>
<tr>
<td>rot</td>
<td>11.11±0.28</td>
<td><strong>10.30±0.27</strong></td>
<td><strong>10.30±0.27</strong></td>
<td><strong>10.29±0.27</strong> (10%)</td>
</tr>
<tr>
<td>bg-rand</td>
<td>14.58±0.31</td>
<td>11.28±0.28</td>
<td><strong>6.73±0.22</strong></td>
<td>10.38±0.27 (40%)</td>
</tr>
<tr>
<td>bg-img</td>
<td>22.61±0.37</td>
<td>23.00±0.37</td>
<td><strong>16.31±0.32</strong></td>
<td><strong>16.68±0.33</strong> (25%)</td>
</tr>
<tr>
<td>rot-bg-img</td>
<td>55.18±0.44</td>
<td>51.93±0.44</td>
<td>47.39±0.44</td>
<td><strong>44.49±0.44</strong> (25%)</td>
</tr>
<tr>
<td>rect</td>
<td><strong>2.15±0.13</strong></td>
<td>2.41±0.13</td>
<td>2.60±0.14</td>
<td><strong>1.99±0.12</strong> (10%)</td>
</tr>
<tr>
<td>rect-img</td>
<td>24.04±0.37</td>
<td>24.05±0.37</td>
<td>22.50±0.37</td>
<td><strong>21.59±0.36</strong> (25%)</td>
</tr>
<tr>
<td>convex</td>
<td>19.13±0.34</td>
<td><strong>18.41±0.34</strong></td>
<td><strong>18.63±0.34</strong></td>
<td><strong>19.06±0.34</strong> (10%)</td>
</tr>
</tbody>
</table>