Deep learning of representations and its application to computer vision

Ian Goodfellow
Summary

• Deep learning background

• Four articles:
  • Spike-and-slab modeling
  • Multi-prediction deep Boltzmann machines
  • Maxout
  • Street number transcription
Machine learning

“A computer program is said to learn from experience \( E \) with respect to some class of tasks \( T \) and performance measure \( P \), if its performance at tasks in \( T \), as measured by \( P \), improves with experience \( E \).”

-Tom Mitchell
Maximum likelihood estimation

- Pick parameters that maximize model’s probability of generating the observed data
- Given enough data, recovers the true model
Gradient descent

Global minimum at $x=0$. Since $f'(x)=0$, gradient descent halts here.

For $x<0$, we have $f'(x)<0$, so we can decrease $f$ by moving rightward.

For $x>0$, we have $f'(x)>0$, so we can decrease $f$ by moving leftward.

$f(x) = \frac{1}{2}x^2$

$f'(x) = x$
Supervised Learning

- Data is features $X$ and targets $y$
- Goal: learn to map $x$ to $y$
- Classification: discrete $y$
- Regression: continuous $y$
Unsupervised learning for feature learning

Original space

- × × positive examples
- • • negative examples

\( x_1 \) vs. \( x_2 \)

\( \phi \)-mapped space

- × × positive examples
- • • negative examples

\( \phi(x)_1 \) vs. \( \phi(x)_2 \)
Deep learning

Our reconstructions are best viewed in electronic form. Visualizing and Understanding Convolutional Networks

Figure 2.

(a): 1st layer features without feature scale clipping. Note that one feature dominates. (b): 1st layer features for "pomeranian" drops significantly. (c): a visualization of this feature map projected down into the input image (black square), along with visualizations of this map from other images. The first row example shows the strongest feature to be the text on the car. When this is covered-up the activity in the feature map decreases (blue area in (b)). (d): a map of correct class probability, as a function of the position of the gray square. E.g. when the dog's face is obscured, the probability for "pomeranian" drops significantly. (e): the most probable label as a function of occluder position. E.g. in the 1st row, column) and see how the top (layer 5) feature maps ((b) & (c)) and classifier output ((d) & (e)) changes. (b): for each layer, probability of correct class as a function of the occluder position. For most locations it is "pomeranian", but if the dog's face is obscured but not the ball, then it predicts "tennis ball". In the 3rd example contains multiple objects. The strongest feature in layer 5 picks out the faces, but the classifier is sensitive to the wheel. The 2nd example, text on the car is the strongest feature in layer 5, but the classifier is most sensitive to the wheel. The 4th example is the Vizsla. Here, we systematically cover up different parts of the image. In the 2nd row, text on the car is the strongest feature in layer 5, but the classifier is sensitive to the ball. The 3rd row contains a dog with a neck brace. Here, the classifier is sensitive to the neck brace. (i) the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of object parts. (f) The feature maps from the 3rd hidden layer. (g) The feature maps from the 2nd hidden layer. (h) The feature maps from the 1st hidden layer. (i) The visible layer, with input pixels for each image. The feature maps from the visible layer show more distinctive features and fewer "dead" features. (j) A visualization of feature map projections from the 1st hidden layer. (k) A visualization of feature map projections from the 2nd hidden layer. (l) A visualization of feature map projections from the 3rd hidden layer. (m) A visualization of feature map projections from the visible layer. (n) A visualization of feature map projections from the output layer. (o) A visualization of feature map projections from the true label layer. (p) A visualization of feature map projections from the predicted label layer. (q) A visualization of feature map projections from the error layer. (r) A visualization of feature map projections from the activity layer. (s) A visualization of feature map projections from the classification layer. (t) A visualization of feature map projections from the object identity layer. (u) A visualization of feature map projections from the object parts layer. (v) A visualization of feature map projections from the object components layer. (w) A visualization of feature map projections from the object contours layer. (x) A visualization of feature map projections from the object edges layer. (y) A visualization of feature map projections from the object corners layer. (z) A visualization of feature map projections from the object layers. (aa) A visualization of feature map projections from the object parts layers. (bb) A visualization of feature map projections from the object parts layers. (cc) A visualization of feature map projections from the object parts layers. (dd) A visualization of feature map projections from the object parts layers. (ee) A visualization of feature map projections from the object parts layers. (ff) A visualization of feature map projections from the object parts layers. (gg) A visualization of feature map projections from the object parts layers. (hh) A visualization of feature map projections from the object parts layers. (ii) A visualization of feature map projections from the object parts layers. (jj) A visualization of feature map projections from the object parts layers. (kk) A visualization of feature map projections from the object parts layers. (ll) A visualization of feature map projections from the object parts layers. (mm) A visualization of feature map projections from the object parts layers. (nn) A visualization of feature map projections from the object parts layers. (oo) A visualization of feature map projections from the object parts layers. (pp) A visualization of feature map projections from the object parts layers. (qq) A visualization of feature map projections from the object parts layers. (rr) A visualization of feature map projections from the object parts layers. (ss) A visualization of feature map projections from the object parts layers. (tt) A visualization of feature map projections from the object parts layers. (uu) A visualization of feature map projections from the object parts layers. (vv) A visualization of feature map projections from the object parts layers. (ww) A visualization of feature map projections from the object parts layers. (xx) A visualization of feature map projections from the object parts layers. (yy) A visualization of feature map projections from the object parts layers. (zz) A visualization of feature map projections from the object parts layers. (aaa) A visualization of feature map projections from the object parts layers. (bbb) A visualization of feature map projections from the object parts layers. (ccc) A visualization of feature map projections from the object parts layers. (ddd) A visualization of feature map projections from the object parts layers. (eee) A visualization of feature map projections from the object parts layers. (fff) A visualization of feature map projections from the object parts layers. (ggg) A visualization of feature map projections from the object parts layers. (hhh) A visualization of feature map projections from the object parts layers. (iii) A visualization of feature map projections from the object parts layers. (jjj) A visualization of feature map projections from the object parts layers. (kkk) A visualization of feature map projections from the object parts layers. (lll) A visualization of feature map projections from the object parts layers. (mmm) A visualization of feature map projections from the object parts layers. (nnn) A visualization of feature map projections from the object parts layers. (ooo) A visualization of feature map projections from the object parts layers. (ppp) A visualization of feature map projections from the object parts layers. (qqq) A visualization of feature map projections from the object parts layers. (rrr) A visualization of feature map projections from the object parts layers. (sss) A visualization of feature map projections from the object parts layers. (ttt) A visualization of feature map projections from the object parts layers. (uuu) A visualization of feature map projections from the object parts layers. (vvv) A visualization of feature map projections from the object parts layers. (www) A visualization of feature map projections from the object parts layers. (xxx) A visualization of feature map projections from the object parts layers. (yyy) A visualization of feature map projections from the object parts layers. (zzz) A visualization of feature map projections from the object parts layers. (aaa) A visualization of feature map projections from the object parts layers.
Spike-and-Slab Sparse Coding

- Co-authors: Aaron Courville and Yoshua Bengio
- Motivated by Adam Coates’ work on feature learning and feature extraction
- Faster form of variational inference
- Component of a deep model
Motivating CIFAR-10 results

- Validation set accuracy (Coates and Ng 2011):
  - RBM features encoded with RBM: 74.1%
  - RBM features encoded with sparse coding: 76.7%
- Test set accuracy (Courville et al 2011):
  - ssRBM: 76.7%
Variational learning

- Approximate intractable \( P(h|v) \) with tractable \( Q(h) \)
- Use \( Q \) to construct a lower bound on the log likelihood
Variational inference

- $\mathcal{L}(v, Q) = \mathbb{E}_{h \sim Q}[\log P(v, h)] + H_Q(h)$
- Maximizing this corresponds to minimizing $D_{KL}(Q(h) \| P(h \mid v))$
- Often requires both analytical and iterative optimization
The Spike-and-Slab Sparse Coding (S3C) Generative Model

\[ p(h_i = 1) = \sigma(b_i) \]
\[ p(s_i \mid h_i) = \mathcal{N}(s_i \mid h_i \mu_i, \alpha_{ii}^{-1}) \]
\[ p(v_d \mid s, h) = \mathcal{N}(v_d \mid W_d : (h \circ s), \beta_{dd}^{-1}) \]
Scaling beyond previous work

Scale of Unsupervised Learning

Number of Latent Variables vs. Number of Training Patches

Spike-and-Slab work: Mohammed et al., 2011; Zhou et al., 2009; Garrigues and Olsahusen, 2008; Lücke and Sheik, 2011; Titsias and Lázaro-Gredilla, 2011
Scaling to more classes and fewer labeled examples per class

- Clic
CIFAR-100 Results

- OMP-1+3
- SC+3
- S3C+3
- S3C+P
- OMP-1+L

OMP-1+L: Jia and Huang 2011
Transfer Learning Challenge

Labeled training set:

Self-taught learning with S3C won the challenge with a test set accuracy of 48.26%
Multi-Prediction deep Boltzmann machines

- Co-authors: Mehdi Mirza, Aaron Courville, Yoshua Bengio
- Simplified training procedure for deep Boltzmann machines
- Improved accuracy of approximate inference
Typical DBM Training

1. Greedy layerwise pretraining

2. Joint generative training

3. Discriminative fine-tuning

Salakhutdinov and Hinton, 2009
Sampling-based approximations

- $p(x; \theta) = \exp(-E(x; \theta))/Z(\theta)$
- What if $Z(\theta)$ is intractable?
  - $\frac{\partial}{\partial \theta_i} \log Z = -\mathbb{E}_x \left[ \frac{\partial}{\partial \theta_i} E(x; \theta) \right]$
- Approximate expectations via sampling
- CD-k: sample k steps from data points
- SML/PCD: sample continuously, use low learning rate
## Simplify, simplify, simplify

<table>
<thead>
<tr>
<th></th>
<th>Classic approach</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td># models</td>
<td>#layers+2</td>
<td>1</td>
</tr>
<tr>
<td># criteria</td>
<td>#layers+2</td>
<td>1</td>
</tr>
<tr>
<td>Classifier</td>
<td>Extra classifier model</td>
<td>Same unified probabilistic model</td>
</tr>
</tbody>
</table>
Multi-Prediction Training

Randomly sample different inference problems

Backprop through the mean field inference graph
Benefits of Multi-Prediction Training

• Learning rate doesn’t affect approximation accuracy

• Training compensates for approximate inference

• Similar to Stoyanov et al 2011
Multi-Inference Trick

**Mean Field Iteration**

Previous State → Step 1 → Step 2

**Multi-Inference Iteration**

Previous State + Reconstruction → Step 1 → Step 2

The graph shows the comparison of Mean field inference and Multi-inference over training epochs.
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Test error with fine-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S&amp;H 2009*</td>
<td>0.95</td>
</tr>
<tr>
<td>Centered DBM</td>
<td>1.22</td>
</tr>
<tr>
<td>MP-DBM</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Centering: Montavon and Müller, 2012

Multi-prediction, 2X hidden units, no fine-tuning*: 0.91

*Retrained using validation set.
<table>
<thead>
<tr>
<th>Classifier</th>
<th>Classic approach</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td># models</td>
<td>#layers+2</td>
<td>1 ✓</td>
</tr>
<tr>
<td># criteria</td>
<td>#layers+2</td>
<td>1 ✓</td>
</tr>
<tr>
<td>Classifier</td>
<td>Extra classifier model</td>
<td>Same unified probabilistic model</td>
</tr>
</tbody>
</table>
Maxout Networks
by Ian Goodfellow
Joint work with
David Warde-Farley    Mehdi Mirza    Aaron Courville    Yoshua Bengio

with acknowledgments to
Frédéric Bastien    Yann Dauphin    Pascal Lamblin
Traditional activation functions

$h_1$, $h_2$, $Z_1$, $Z_2$, $V_1$, $V_2$, $V_3$

Activation

“Weight vector”

“Filter”

Weight
Logistic sigmoid activation function

Output of neuron

Input to neuron
The vanishing gradient problem

Logistic sigmoid activation function

Rectified linear activation function
Uh-oh

Rectified linear activation function
$h_i = \max_j z_{ij}$
Comparing maxout to rectifiers
Effectiveness of pooling

Comparison of large rectifier networks to maxout

- Maxout
- Rectifier, no channel pooling
- Rectifier + channel pooling
- Large rectifier, no channel pooling

Figure 6. Effectiveness of pooling

Figure 7. Comparison of large rectifier networks to maxout
Applications of maxout

• Multiplayer game matchmaking: Laufer et al, 2013
• Text detection: Jaderberg et al 2014
• Text transcription: Alsharif and Pineau, 2013
• Simplifying optimization: Gulcehre 2013
• Recurrent networks: Pascanu 2014
• Whale call detection: Smirnov 2013
• Black-box classification: Xie et al, 2013
Street number transcription

- Co-authors: Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet

- Use convolutional networks to read address numbers from Street View Images

- Automated transcription of over 100 million real address numbers
Want

Neural net → 243
Neural net → 43
Neural net → 143
Proposed architecture:
- **end-to-end** learning
- **no explicit** segmentation
- **integrated** character recognition
- no need for a baseline
- no per-character GT required
- output entire sequence **at once**

6 softmax:
- 1 for length
- 1 per character
Training

Log likelihood:

\[
\log P(L = l \mid X) + \sum_{i=1}^{L} \log P(S_i = s_i \mid X)
\]
\[
\arg\max_{L, S_1, \ldots, S_L} \log P(S \mid X)
\]

\[
\log P(L = 1) + \log P(S_1 = \text{"1"})
\]

\[
\log P(L = 2) + \log P(S_{1:2} = \text{"17"})
\]

\[
\log P(L = 3) + \log P(S_{1:3} = \text{"175"})
\]

\[
\log P(L = 4) + \log P(S_{1:4} = \text{"1751"})
\]
<table>
<thead>
<tr>
<th></th>
<th>Coverage@ human accuracy (98%)</th>
<th>Accuracy</th>
<th>Per Character Accuracy</th>
<th>Per Character Accuracy (Prev. state of the Art)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public SVHN</td>
<td>95.6%</td>
<td>96%</td>
<td>97.8%</td>
<td>97.7%</td>
</tr>
<tr>
<td>Private Dataset</td>
<td>89%</td>
<td>91%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example failures:

1180 vs. 1780

1844 vs. 184

2 vs 239

100 vs. 676
Effect of depth

Accuracy versus depth

Sequence transcription accuracy

Number of hidden layers

One fully connected layer
Two fully connected layers
Effect of # of parameters

Effect of model size

Accuracy

Number of parameters

- 3-layer model, varying conv. layer sizes
- 3-layer model, varying fully connected layer size
- 5-layer model
- 11-layer model, varying conv. layer sizes
Conclusion

- Unsupervised learning useful when very little labeled data available
- Generative models useful for missing value problems
- Implicit ensembles and/or lots of data are much more effective