Neural Networks: Deep Learning (2)

Lyle Ungar

Multilevel network: architecture, link functions
CNNs: local receptive fields, max pooling

-----------------------

Regularization: $L_2$, early stopping, dropout
Gradient Descent (again)
Semi-supervised and transfer learning
Modern deep nets

- Often use rectified linear units (ReLUs)
  - Faster, less problems of saturation than logistic
- Use a variety of loss functions
  - Log-likelihood with softmax
- Can be very deep
- Solved with mini-batch gradient descent
- Regularized using $L_2$ penalty plus “dropout”
  - and partial convergence and ..

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

for $j = 1, \ldots, K$. 

$$f(x) = \max(0, x)$$
Regularization

- $L_2$ and/or $L_1$
- Early stopping
- Max norm ($L_\infty$)
  - Weight clipping
  - Gradient clipping
- Dropout

Dropout

- Randomly (temporarily) remove a fraction $p$ of the nodes (with replacement)
  - Usually $p = 1/2$

- Repeatedly doing this samples (in theory) over exponentially many networks
  - Bounces the network out of local minima

- For the final network
  use all the weights
  but shrink them by $p$
Gradient descent

- **Gradient descent**
  - Minibatch
  - Gradient clipping

- **Momentum**
  \[
  \Delta w^t = \eta \frac{\delta Err}{\delta w} + m \Delta w^{t-1}
  \]

- **Learning rate adaptation**
  - Adagrad and friends
Learning rate adaption

- Adjust the learning rate over time

\[ \Delta w_t^t = \eta(t) \frac{\delta Err}{\delta w} \]

- Adagrad: make the learning rate depend on previous changes in each weight
  - increases the learning rate for more sparse parameters and decreases the learning rate for less sparse ones.

\[ \Delta w_j^t > \frac{\eta}{||\delta w_j^t||_2} \frac{\delta Err}{\delta w_j} \]
Learning rate

[Image: Graph showing loss vs. epoch with curves for different learning rates: very high, low, high, and good learning rates. The graph also shows a plot of loss vs. epoch with a blue line indicating a stable learning process.]

Feature Scaling (standardizing)

- **Idea:** Ensure that feature have similar scales
- **Makes gradient descent converge much faster**

Is deep learning scale invariant?
Lots of fancy network structures

Convolutional (different sizes)
Or fully connected
Maxpool
Concatenation
Softmax

Some layers use dropout
googlenet
Validation classification

<table>
<thead>
<tr>
<th>mite</th>
<th>container ship</th>
<th>motor scooter</th>
<th>leopard</th>
</tr>
</thead>
<tbody>
<tr>
<td>black widow</td>
<td>lifeboat</td>
<td>go-kart</td>
<td>jaguar</td>
</tr>
<tr>
<td>cockroach</td>
<td>amphibian</td>
<td>moped</td>
<td>cheetah</td>
</tr>
<tr>
<td>tick</td>
<td>fireboat</td>
<td>bumper car</td>
<td>snow leopard</td>
</tr>
<tr>
<td>starfish</td>
<td>drilling platform</td>
<td>golfcart</td>
<td>Egyptian cat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>grille</th>
<th>mushroom</th>
<th>cherry</th>
<th>Madagascar cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>convertible</td>
<td>agaric</td>
<td>dalmatian</td>
<td>squirrel monkey</td>
</tr>
<tr>
<td>grille</td>
<td>mushroom</td>
<td>grape</td>
<td>spider monkey</td>
</tr>
<tr>
<td>pickup</td>
<td>jelly fungus</td>
<td>elderberry</td>
<td>titi</td>
</tr>
<tr>
<td>beach wagon</td>
<td>gill fungus</td>
<td>currant</td>
<td>indri</td>
</tr>
<tr>
<td>fire engine</td>
<td>dead-man's-fingers</td>
<td>ffordshire bullterrier</td>
<td>howler monkey</td>
</tr>
</tbody>
</table>
Validation classification
Validation localizations
Retrieval experiments

First column contains query images from ILSVRC-2010 test set, remaining columns contain retrieved images from training set.
Now used for image search; Benefit: Good Generalization

Both recognized as “meal”

Jeff Dean, google
Sensible Errors (sometimes)

“snake”

“dog”

Jeff Dean, google
Now used for image search

Works in practice… for real users

Wow.
The new Google plus photo search is a bit insane.
I didn’t tag those... :)
Now used for image search

Works in practice... for real users

Google Plus photo search is awesome. Searched with keyword 'Drawing' to find all my scribbles at once :D

Jeff Dean, google
Questions?
Semi-supervised learning

- Use unlabeled data $X_0$ to derive new features $z = \phi(x)$
- Train $y = f(\phi(x), w)$
Transfer learning

- Use one data set \((X_0, y_0)\) to train a model
- Find feature transformations \(\phi(x)\)
- Use those transformations \(\phi(x)\) on data from data set with a different label, \(y\).
Transfer learning for NNets

Data set for learning $\phi(x)$

Data set with target $y$.
New AI can guess whether you're gay or straight from a photograph

An algorithm deduced the sexuality of people on a dating site with up to 91% accuracy, raising tricky ethical questions
Deep neural networks are more accurate than humans at detecting sexual orientation from facial images.

Michal Kosinski & Yilun Wang

We show that faces contain much more information about sexual orientation than can be perceived and interpreted by the human brain. We used deep neural networks to extract features from 35,326 facial images. These features were entered into a logistic regression aimed at classifying sexual orientation. Given a single facial image, a classifier could correctly distinguish between gay and heterosexual men in 81% of cases, and in 74% of cases for women. Human judges achieved much lower accuracy: 61% for men and 54% for women. The accuracy of the algorithm increased to 91% and 83%, respectively, given five facial images per person. Facial features employed by the classifier included both fixed (e.g., nose shape) and transient facial features (e.g., grooming style). Consistent with the prenatal hormone theory of sexual orientation, gay men and women tended to have gender-atypical facial morphology, expression, and grooming styles. Additionally, given that companies and governments are increasingly using computer vision algorithms to detect people’s intimate traits, our findings expose a threat to the privacy and safety of gay men and women.

https://osf.io/fk3xr/ 2017
Deep learning case study

- Download images and labels from a dating site
  - where people declare their sexual orientation
- Only keep images with a single “good” face
  - Use Face++ to identify faces -- yielded 35,000 faces
- Use M-turkers to QC & restrict to Caucasians
- Use pretrained CNN to compute ~ 4,000 ‘scores’/image
  - VGG-Face was trained on 2.6 million faces
- Use logistic regression on SVD of the 4,000 scores
  - report cross-validation error predicting gay/straight
What you should know

- CNN
  - local receptive field, max pooling
- Rectified Linear Unit (ReLU)
- Dropout
- Back-propagation, momentum,
- Mini-batch
- At least four kinds of regularization
- Transfer learning
Questions?