Convolutional Networks and Applications in Vision

Sai Kumar Kayala
What is CNN?

- Type of Deep Neural Network
- Inspired by the organization of Visual Cortex
- Typically takes a image as Input
- Classifies a class as Output
- Forms feature maps using filters
- Has the ability to learn these filters
Structure of a CNN
Computations in a CNN

Conv-Layer/Filter Bank Layer

Non-Linearity Layer

Pooling Layer

$$y_j = b_j + \sum_i k_{i,j} \ast x_i$$

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

$$f(x) = \begin{cases} 
0 & \text{for } x < 0 \\
 x & \text{for } x \geq 0 
\end{cases}$$
Advantages of CNN

Best Feature Learning!

Computationally Efficient

Weight Sharing

Better performance than Densely connected Neural Network
Computational Efficiency

For Fully Connected:
A 28x28 image
250 nodes hidden layer would need
250*784+1=19601 parameters

For CNN:
With 64 filters of size 5
We need 64*25=1600 parameters
Supervised-Learning

We have a labelled data.

Forward step:
Predict output given an input

Back propagation:
Calculate ‘Gradients’ based on truth and prediction
Using this gradient we adjust the “Kernels”
Gradient descent algorithm is used which ensures we reach minima (may be global or local)
Unsupervised-Learning

Having lots of labelled data is hard!

Make use of unlabelled data using sparse coding algorithms

Use little available labelled data to fine tune.

Predictive Sparse Decomposition can be used to train single stage of the network.

Under a simple assumption that similar inputs would lead to similar activations.
Learning with Predictive Sparse Decomposition

\[ E(Z, W, K) = \| X - WZ \|^2_2 + \lambda \| Z \|^1_1 + \| Z - C(X, K) \|^2_2 \]

- **X** - Input
- **W** - Dictionary element (Filter bank)
- **Z** - Feature Map
- Goal is to find **Z** that minimizes **E**
Results

Caltech - 101 dataset:  2-stage network
101 categories   1st stage - 9x9 64 filters
About 50 images per category   2nd stage - 256 x 16 feature maps

### Average recognition rates on Caltech-101.

<table>
<thead>
<tr>
<th></th>
<th>$R_{abs} - N - P_A$</th>
<th>$R_{abs} - P_A$</th>
<th>$N - P_M$</th>
<th>$P_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U^+$</td>
<td>65.5%</td>
<td>60.5%</td>
<td>61.0%</td>
<td>32.0%</td>
</tr>
<tr>
<td>$R^+$</td>
<td>64.7%</td>
<td>59.5%</td>
<td>60.0%</td>
<td>29.7%</td>
</tr>
<tr>
<td>$U$</td>
<td>63.7%</td>
<td>46.7%</td>
<td>56.0%</td>
<td>9.1%</td>
</tr>
<tr>
<td>$R$</td>
<td>62.9%</td>
<td>33.7%</td>
<td>37.6%</td>
<td>8.8%</td>
</tr>
</tbody>
</table>
Few Applications

Object recognition  OCR  Face Detection

Segmentation

Images showing applications of object recognition, OCR, and face detection, with examples of text and images being processed.
CNN’s in Medicine

Pneumonia Detection

Personalized Medicine

Heart Failure Prediction

Cancer detection
Generative Networks

DCGAN

Face Generator

Deep Dream
Natural Language Processing

We can use a word embedding like glove

A 10 word sentence with 100 dimensional Embedding makes a 10x100 ‘image’

We can use this to extract more features

We can even use a spectrogram of word

![Wave Net](image.png)

Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Char2wav spectrogram output
AlphaGo, OpenAI

Deep Q Learning algorithm

\[ \text{New } Q(s, a) = Q(s, a) + \alpha [R(s, a) + \gamma \max Q'(s', a') - Q(s, a)] \]
DeepFakes

Face App

Few Controversial Applications
Thank you & Questions