Convolutional Neural Networks

Shrey Gupta

Applied Machine Learning (HOUSECS 59-01), Duke University

November 7, 2018

Problems

- **Classification:** labeling an image from a set of classes.
- **Localization:** locating (e.g. via a bounding box) a single labeled object in an image.
- **Object detection:** locating (e.g. via a bounding box) labeled objects in an image.
- **Landmark detection:** detecting specific features (landmarks) in an image.

Classification

CAT

- Task: labeling an image from a set of classes.

Image source: Arthur Ouaknine

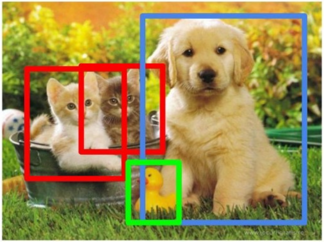
Localization

CAT

- Task: locating (e.g. via a bounding box) a single labeled object in an image.

Image source: Arthur Ouaknine

Object Detection



CAT, DOG, DUCK

- Task: locating (e.g. via a bounding box) labeled objects in an image.

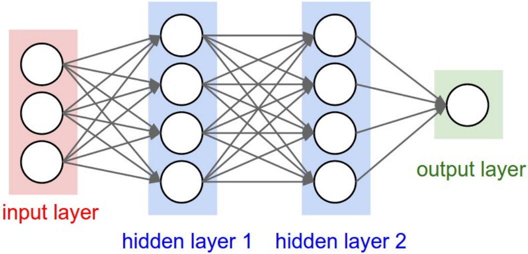
Image source: Arthur Ouaknine

Landmark Detection

- Task: detecting specific features (landmarks) in an image.

Image source: Andrew Ng

Recall



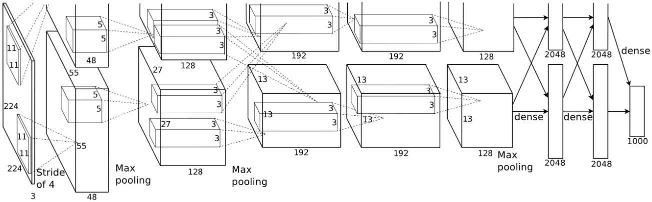
- Input layer: single vector of feature inputs.
- Hidden layer(s): sets of neurons with nonlinear activation, fully connected by weights (with biases) to other layers.
- Output layer: single vector of output scores.

Image source: Stanford University

Issues

- Doesn't scale well to large features, such as images.
- Speech and images are rich in structure (e.g. hierarchy). In the case of images, how can we utilize the structure to our advantage?
- Take advantage of the properties of images to create a new network.

Architecture Summary



- Architecture: convolutional layers, pooling layers, and fully-connected layers.
- Note: consider intensity (for black-and-white images) or RGB values over pixels as inputs.

Image source: Conference on Neural Information Processing Systems

Templates

- Intuitively, we might think of creating *template* images for each class, and using some similarity measure (e.g. dot product) to match to data.
- This is exactly what we'll do, but through a "hierarchical" approach.

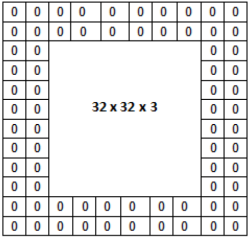
Convolutional Layers

- Recall: neurons match inputs to patterns.
- Create $w \times h$ templates for $w \times h$ receptive fields centered at (x, y) locations in the image, with stride s (usually small) along each dimension.
  - Each receptive field is hence *local*.
- Match the template to the field, and compute a score via the *dot product*.
- Construct a new image of the activation of the scores.

Stride

- For each template, there are several receptive fields in the image that we can observe.
- Beginning at the top left receptive field, the next receptive field will be stride s to the right and/or bottom.

Aside: Padding



- Performing a convolution with an image creates another image that will be smaller than the original one.
- To rectify this potential issue, *pad* the edges of the image with zeros.

Image source: Adit Deshpande

Volume

- A given layer may contain a *stack* of images.
  - Ex: a color (RGB) image is actually a stack of three images.
- The template is convolved with each image in the stack, and then summed into a single image.
- We might create several templates, creating volume in the next layer, since each template creates a new image.
  - This allows the algorithm to learn multiple features at the same hierarchical level.

Pooling Layers

- Reduces the number of parameters (and hence computations).
- Controls overfitting: intuitively, allows for translational invariance.
- Compute the maximum score for each $w \times h$ subimage across the image (overlap possible).
- Construct a new, smaller image of the maximum scores.
- Common variations of the sizes include 2×2 and 2×3 (with overlapping).

Fully-Connected Layers

- These layers are identical to the hidden layers as before: sets of neurons with nonlinear activation (usually ReLU), fully connected by weights (with biases) to other layers.
- Input from the final convolutional layer, consisting of transformed and smaller features.

Aside: Convolution

- Why are these called convolutional neural networks?
- Recall the convolutional layers in the network, which perform template-matching; with stride 1, this can be viewed as a convolution over the image.

Backpropagation

- Backpropagation is performed in a manner identical to fully-connected neural networks (i.e. as before).

Example

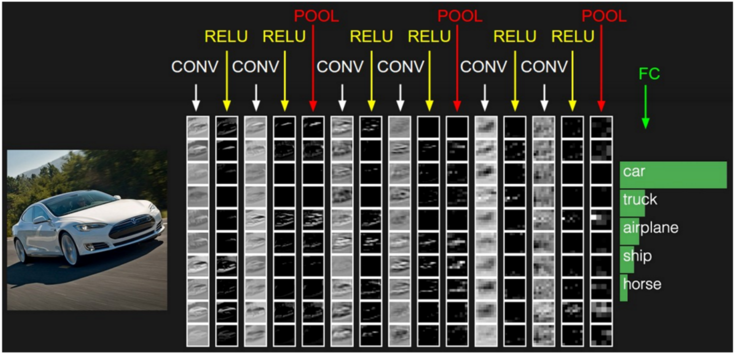


Image source: Stanford University

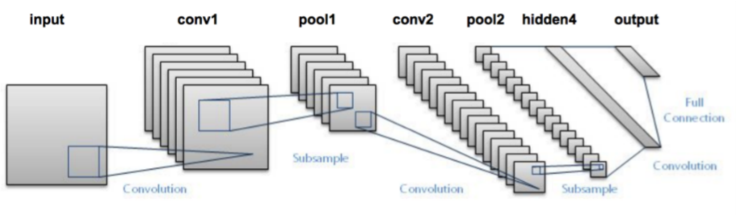
Layer Representation

- The templates at each layer of the network represent understanding different features of the image.
- For example, earlier layers might look at edges and outlines.

Example Networks

- The following slides explore various convolutional neural networks over the past several years.
- Each network presented crucial insight into the construction of CNNs, and may be valuable to you when designing or choosing architectures.

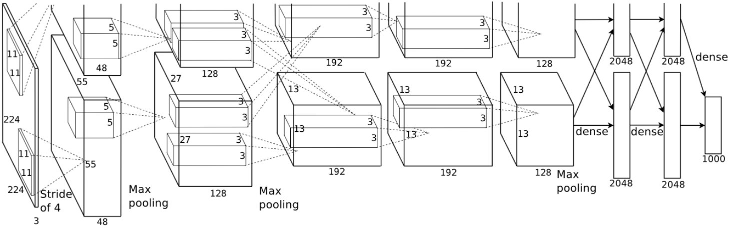
LeNet



- Yann LeCun developed LeNet in 1998, a convolutional neural network that classifies handwritten digits.

Image source: Adrian Rosebrock

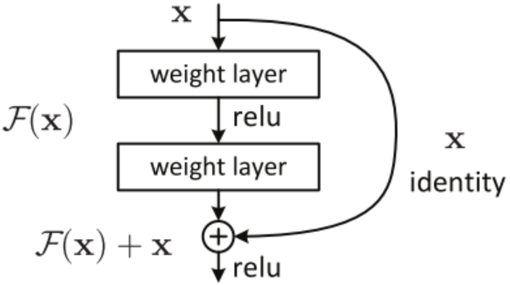
AlexNet



- Trained on ImageNet, AlexNet made an important step forward in image classification with great improvements in performance as a *deep* CNN.

Image source: Conference on Neural Information Processing Systems

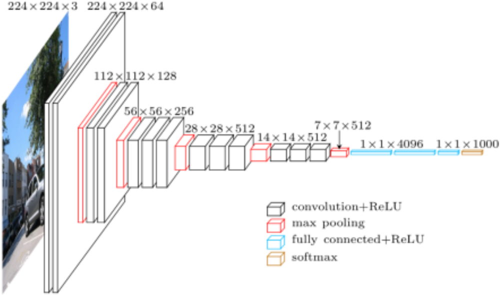
ResNet



- ResNet introduced a new type of connection between neurons to address the *vanishing gradient problem*.

Image source: Vincent Fung

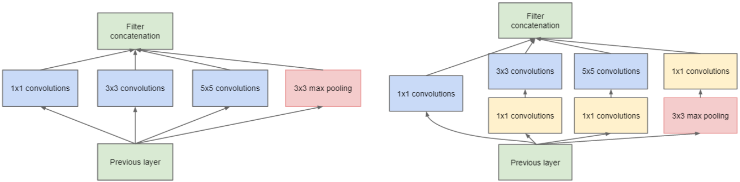
VGGNet



- VGGNet was a simple convolutional neural network that still achieved high performance.

Image source: Adrian Rosebrock

Inception Network

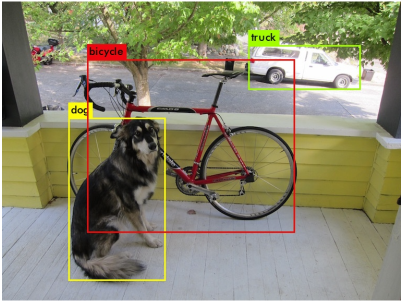


(a) Inception module, naïve version (b) Inception module with dimension reductions

- The inception network allowed the *model* to choose the "optimal" transformations (convolutions) between different layers.

Image source: Joyce Xu

YOLO



- Joseph Redmon invented "You Only Look Once" (YOLO) in 2015, performing real-time object detection with performance higher than ever before.

Image source: Joseph Redmon

Non-maximum Supression

- It's likely that the algorithm will find multiple bounding boxes around the actual instance of an object.
- For a given class and object detection, only choose the bounding box with the maximum score (i.e. supress the others).

Data Augmentation

- Important to include larger sets of training data.
- Ex: horizontal/vertical flips, rotations, resizing, cropping, changes in contrast/brightness, and/or distortions.

Transfer Learning

- Deep convolutional neural networks are very large, and can take very long to train!
- Solution: borrow (in earlier layers) or initialize weights from an open-source model trained on similar or more general data to speed up your model's convergence.

Notebook

- Today's notebook will work through an example of convolutional neural networks.

References

- *Deep Convolutional Neural Nets*
- *Convolutional Neural Networks for Visual Recognition*