Convolutional Neural Network

UW CSE 473
March/4th 2019
Beibin Li
Review
Which Direction

Regression:

Angle = [-540°, 540°]

Classification:

- Turn left
- Turn right
- Stay Still
Which Move

Conv. Module #1

Conv. Module #2

Classification

Design Utility Function (HW 3)

We can train a CNN to replace the manual utility function

Utility = CNN( )
Which Digit?
Which Object?

Homework 5

Image Classification:

- Tree
- Face
- Car
- Dog
- Plane
Homework 5

- How many trainable parameters are in the model?

- What is the best training accuracy?

- What is the best validation accuracy? Is it better than the ones in previous questions?

- According to the training and validation accuracies, does the model overfit your training data?

Note: It is ok if your accuracy is not the same as your friend’s.
CNN

convolution + nonlinearity

max pooling

convolution + pooling layers

fully connected layers

Nx binary classification

bird

P_{bird}

sunset

P_{sunset}

dog

P_{dog}

cat

P_{cat}
Convolution: from fixed to learnable

Expert-Designed Convolution: SIFT, HoG, LBP, ...

Learn Flexible Parameters
FC Activation: From Step Function to Sigmoid

Note: In homework 5, because there is only one FC layer, we do not need nn.Sigmoid()
CNN Activation: From Sigmoid to ReLU

**Sigmoid**

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

**TanH**

\[ \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \]

**ReLU**

\[ f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \]
Representation

Input = (3, width, height)

Meet Softmax

\[ \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K. \]
Inference

Note: In homework 5, you need to reshape the tensor between Convolutional and Fully-Connected Layers
Learning

\[
\begin{align*}
O_{11} &= F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22} \\
O_{12} &= F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23} \\
O_{21} &= F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32} \\
O_{22} &= F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}
\end{align*}
\]

- Details:
  - [https://www.slideshare.net/EdwinEfranJimnezLepe/example-feedforward-backpropagation](https://www.slideshare.net/EdwinEfranJimnezLepe/example-feedforward-backpropagation)
Underfit v.s. Overfit

Regression:

- **M = 1**: predictor too inflexible; cannot capture pattern
- **M = 3**: predictor too flexible; fits noise in the data

Classification:

- **x₁ vs. x₂**: left: simple decision boundary; middle: complex decision boundary; right: intricate decision boundary
Homework 5
Neural Network (Q1)
Convolutional Neural Network (Q2)

Conv
Pool
FC
Cross Entropy
Yellow or Blue?
Color Normalization (Q3)
That is NOT Enough

We Have to Go DEEPER
Deep Convolutional Neural Network (Q4)
Make the Design More Flexible

Input:

[8, 16, 32, "pool"]
Exercise: Input [8, ‘pool’, 16, "pool"]
Data Augmentation (Q5)

Random Affine Transformation
Starter Code
Python 3 and Anaconda

- Python: Created at 1991
- Python 1.0: released at 1994
- Lambda, Map, Reduce, etc
- Exception Handling
- Python 2.0: at 2000
- Garbage collection
- Unicode Support
- Python 3.0: at 2008
- Performance & Design Improvements
- Syntax is different, and not backwards compatible

- (optional if you are already familiar with Python)
- Package management
- Environment control
- Nice IDE (Spyder) and debugger
- Jupyter Notebook
- and more…
Deep Learning Libraries

mxnet
TensorFlow
Theano
scikit-learn
Caffe2
Keras
PyTorch
PyTorch

- Dynamic Graph
- Easy and Flexible
- Popular
- Well maintained
- Compatible with Torch and Caffe
- more...