Lecture 8

Regularization

Many methods have been developed to reduce overfitting during training.

- General idea: overfitting results from many parameters working together to overfit data.

- Some commonly used methods:
  - Weight regularization / penalty:
    \[
    \text{loss} + \alpha \sum_{w} \|w\|^2
    \]
    (bias to L2 weight regularization)

- Dropout: randomly drop activities (nodes) during training, restored at evaluation time.
Convolutional Neural Networks

- CNN flattens image → vector, doesn’t preserve locality.
- CNN preserving 2D structure →
  - learn much smaller “filter” matrix
  - local feature map

Idea: drag filter across image, each filter learns 1 feature
- produce 2D maps of features in image

\[ A \left( \sum_{a,b=1}^{3} x_{i+a,j+b} \cdot w_{ab} + b \right) = x_{i,j}^{(l)} \]
- produces hidden layer which is an image!
Can stack filters & convolutional layer

Input

hidden layers

32x32x3

32x32 x \(N\) filters

"channels"

"colors"

"can stack conv. layers!"

\[
A \left( \sum_{i,j,k} x_{ijk} w_{ijk} + b \right)
\]

Pooling layer - reduce image size drastically

\[
\text{max} (a, b, c, d)
\]

\[
\text{avg} (\ldots)
\]

\[
\frac{a+b+c+d}{4}
\]

\[
\frac{\text{max pooling}}{\text{avg}}
\]

\(wbc + b\)

\(\frac{a+b+c+d}{4}\)

\(\frac{a+b+c+d}{4}\)

\(\frac{a+b+c+d}{4}\)

\(\frac{a+b+c+d}{4}\)
CNN is a DNN w/ massive weight sharing

Conv. layers + policy

"Classic CNN arch."