Feed-forward Neural Networks

\[ f = g^{(3)} \left( \frac{W^{(3)} h^{(2)} + b^{(3)}}{a^{(3)}} \right) \]

\[ h^{(2)} = g^{(2)} \left( W^{(2)} h^{(1)} + b^{(2)} \right) \]

\[ h^{(1)} = g^{(1)} \left( W^{(1)} x + b^{(1)} \right) \]

\[ h_k^{(2)} = g^{(2)} \left( \sum \limits_{\ell} W_{k\ell}^{(2)} h_\ell^{(1)} + b_k^{(2)} \right) \]

Homework: Try plotting for random weights.

Change: scale \( W \), # layers, \( g \), without \( b \)
Fitting params $\theta$ 

Stochastic Gradient Descent on cost $c$

$$\theta \leftarrow \theta - \eta \nabla_{\theta} c(\theta)$$

Cost $c$ describes task
- Match $f$ to one-hot class label
- Match $f$ to strength of concrete
- Match $f$ to pixels of next image

**Side-effect:** learn "hidden" representations $h^{(n)}$, and $E$

$W_{n}^{(n)}$ as image:

This "filter" looks for a type of edge.

$h^{(n)}$ indicates which edges present.

Embeddings:
If $x$ one-hot, $W_{x}^{(n)}$ is vector for choice.

Often set $h^{(n)}$ directly to $W_{x}^{(n)}$.

No point doing $W_{x}^{(n)}$ and applying function.

Vector containing $W^{(n)}$, $b^{(n)}$, $\sum_{n=1}^{L}$ and any other params.

Tomorrow:
- how to compute $\nabla_{\theta} c(\theta)$
gg = @(a) 1./(1 + exp(-a));
X = (-1:0.01:1)'; % Nx1
H1 = gg(X * randn(1, 100)); % Nx100
H2 = gg(H1 * randn(100, 50)); % Nx50
F = gg(H2 * randn(50, 1)); % Nx1
plot(X, F);

gg = @(a) 1./(1 + exp(-a));
X = (-1:0.01:1)'; % Nx1
H1 = gg(X * randn(1, 100) * 10); % Nx100
H2 = gg(H1 * randn(100, 50) * 10); % Nx50
F = gg(H2 * randn(50, 1) * 10); % Nx1
plot(X, F);
Specialized Architectures

ConvNets: learned filters move over whole image
don't relearn in every location
⇒ share parameters

Script pooling (example)

"Bag of words" text classification

\[ e(t) = W \times x(t) = \text{"embedding vector for t''th word"} \]

\[ h = \left( \frac{1}{T} \right) \sum_{t=1}^{T} e(t) \quad \text{"document vector"} \]

\[ f = \text{Softmax} \left( V h \right) \quad \text{embedding dim} \times 1 \]
\[ \text{prob. over classes} \quad \# \text{classes} \times \text{embedding dim} \]
NN don't have a convex cost

No unique optimum
⇒ not convex

Contours of cost

Set $w^{(1)}$ to best values for $w^{(2)}$

- These local optima, don't matter
  → because the functions (predictions) are same.

- Not all optima are equivalent.
Initialization

Must not set all \( W^{(l)} \) to zero

- all hidden extract same feature
- SGD applies same update to all params
- all hidden stay same...

Set randomly \( W_{i,j}^{(l)} \sim N(0,1) \)?

[cf WOF note]

- activation \( W^{(l)} \) is sum of \( D \) random values for each hidden

- typically each term in sum
  \( \sim \pm 1 \) (if data standardized)

- \( W^{T} x + i \sim \pm \sqrt{D} \)

- instead \( w \sim N(0, (\frac{1}{\sqrt{D}})^2) \)

MLP course note, have more ideas.

2019 L20 8