Feed-forward Neural Networks

\[
\begin{align*}
    f &= g^{(3)} \left( W^{(3)} h^{(2)} + b^{(3)} \right) a^{(3)} \\
    h^{(2)} &= g^{(2)} \left( W^{(2)} h^{(1)} + b^{(2)} \right) \\
    h^{(1)} &= g^{(1)} \left( W^{(1)} x + b^{(1)} \right)
\end{align*}
\]

\[x, \text{ inputs}\]

\[h^{(2)}_k = g^{(2)} \left( \sum_{\ell} W^{(2)}_{k\ell} h^{(1)}_\ell + b^{(2)}_k \right)\]

**Homework:** Try plotting for random weights.

**Change:** scale \( W \), # layers, \( g \), without \( b \)
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);
Initialization to zero.

Don't set all weights for the same value

Naive: \( W^{(1)} \sim N(0,1) \)

Use "randn"

[Revise background notes, expectations]

Sum of \( k \) random values

They are typically \( i.i.d. \) each

Typically, sum \( \sim \pm \sqrt{k} \)

Example initialization: \( w \sim N(0, (\frac{1}{\sqrt{k}})^2) \)

MLP: Has more sophisticated ideas.
NN don't have a convex cost

No unique optimum
⇒ not convex.

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

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

- Not all optima are equivalent
Regularization - Early-stopping

Could do L2 regularization
Set $\lambda$ ... cross-validate

Cost

Validation cost

Training cost

Overfitting?

Iteration #

want this model.

(val. error - train error)

is not "overfittingness"
Every $R$ updates:

If val. cost is the smallest I've seen:

Store: weights & val cost.

If val. cost hasn't improved in 20 updates, stop. Return weights we are at last stored.

Learning rates $2^t$ → or reduce 2

Learning rate graph:

- $2^t$ increases
- $t$ increases
Getting gradients - Reverse-mode differentiation

```
\[ C \leftarrow f \leftarrow y \leftarrow L(f, y) \]

\[ a^{(L)} \leftarrow b^{(L)} \leftarrow h^{(L-1)} \]

Strategy:

- For every intermediate \( \theta \)
  
  get \( \overline{\theta} = \frac{\partial C}{\partial \theta} \)

  \( \hat{f} = \frac{\partial C}{\partial f} \)

  \( \overline{a_i^{(L)}} = \frac{\partial C}{\partial a_i^{(L)}} \)

  \( \overline{W_{ij}}^{(L)} = \frac{\partial C}{\partial W_{ij}^{(L)}} \)
```