Learning from multiple tasks

Transfer learning

deeplearning.ai

Transfer learning

image recognition (x,y)
↑
pre-training

→ (x,y) — fine-tuning
radiology image ↗   ↖ diagnoses

$\hat{y}$

$w^{[L]}, b^{[L]}$

image recognition
→ 1,000,000   100 ←
↓
→ radiology diagnosis
→ 100   1000

Speech recognition
10h   10,000h

wakeword/triggerword
detection   1h
50h

x
audio

$\hat{y}$

Andrew Ng
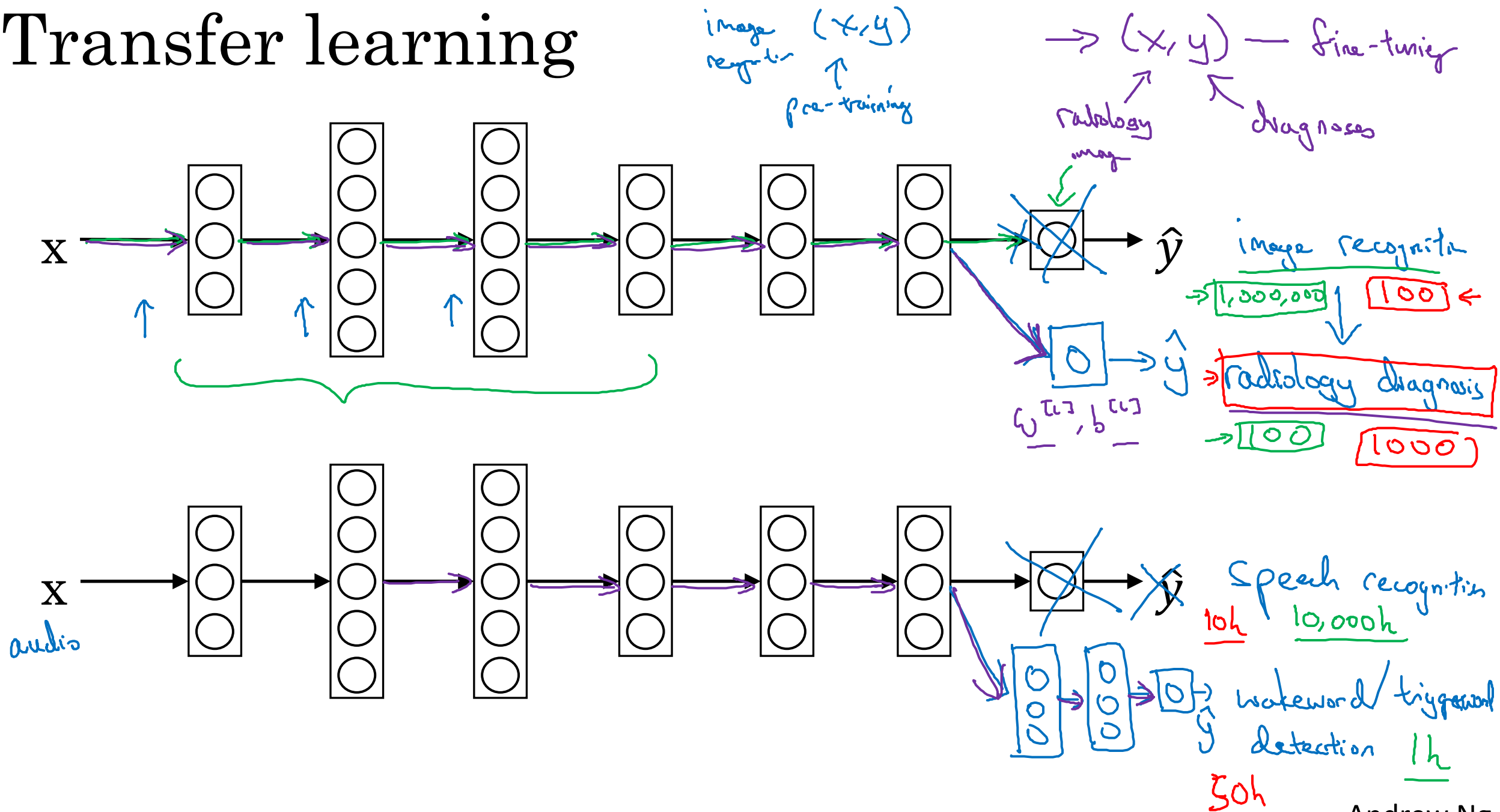
When transfer learning makes sense

Transfer from A → B

- Task A and B have the same input x.

- You have a lot more data for Task A than Task B.

- Low level features from A could be helpful for learning B.

Andrew Ng

deeplearning.ai

Learning from multiple tasks

Multi-task learning

Simplified autonomous driving example

$X^{(i)}$

Pedestrians

Cars

Stop signs
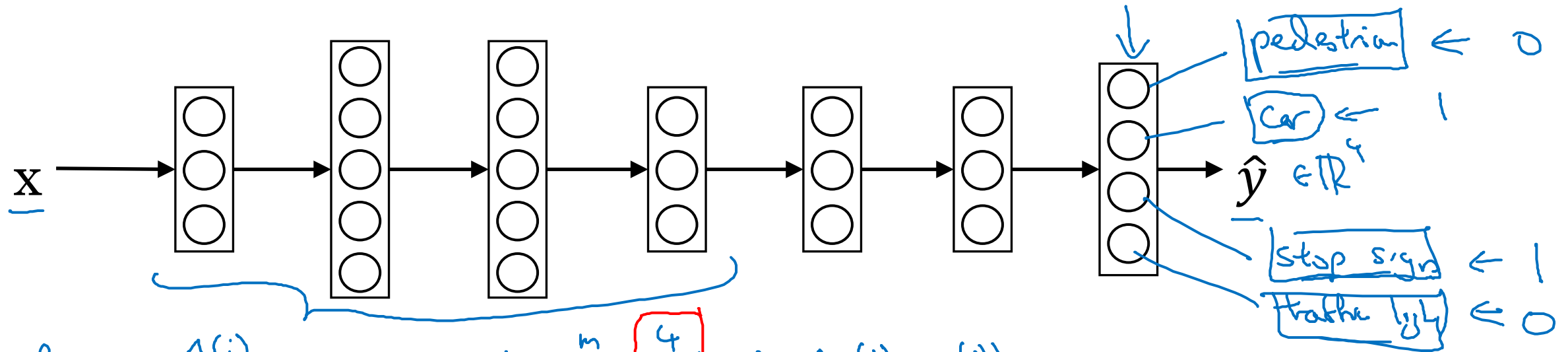
Traffic lights
$\vdots$

$y^{(i)}$

$\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \end{matrix}$

$(4,1)$

$$Y = \begin{bmatrix} | & | & | & & | \\ y^{(1)} & y^{(2)} & y^{(3)} & \cdots, & y^{(m)} \\ | & | & | & & | \end{bmatrix}$$

$(4, m)$

Andrew Ng

Neural network architecture

pedestrian ← 0

Car ← 1

$\hat{y} \in \mathbb{R}^4$

stop sign ← 1

traffic light ← 0

Loss: $\overset{\wedge (i)}{Y}_{(4,1)}$

$\rightarrow \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{4} \mathcal{L}(\hat{y}_j^{(i)}, y_j^{(i)})$

Sum only over value of j with 0/1 label.

↳ Usual logistic loss

$-y_j^{(i)} \log \hat{y}_j^{(i)} - (1 - y_j^{(i)}) \log (1 - \hat{y}_j^{(i)})$

Multi-task learning ←

Unlike softmax regression:

One image can have multiple labels

$Y = \begin{bmatrix} 1 & 1 & 0 & ? \\ 0 & 1 & 0 & 1 \\ ? & ? & 1 & ? \\ ? & ? & 0 & ? \end{bmatrix}$ ←

Andrew Ng

When multi-task learning makes sense

- Training on a set of tasks that could benefit from having shared lower-level features.
- Usually: Amount of data you have for each task is quite similar.

A   1,000,000

↓   ↓

B   1,000

A_1 1,000
A_2 1,000
⋮   ⋮   99,000

A_{100} 1,000
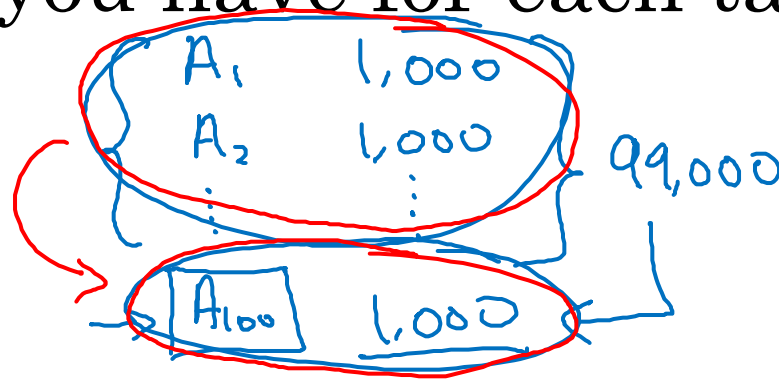
- Can train a big enough neural network to do well on all the tasks.