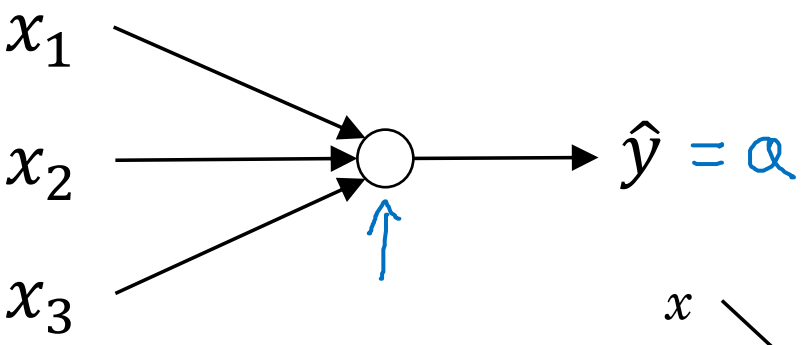
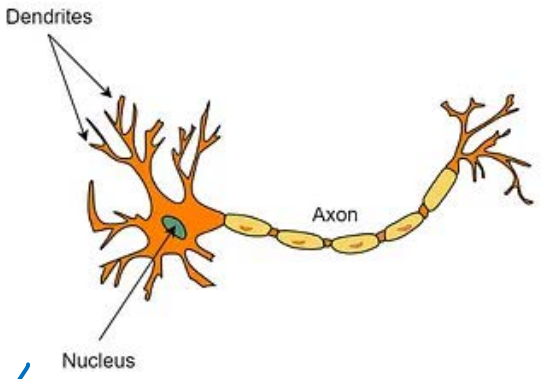
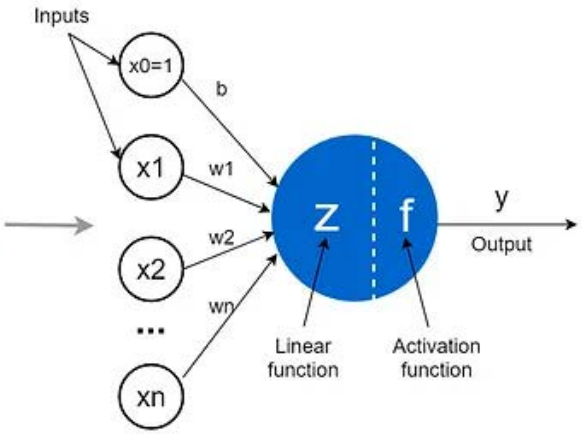
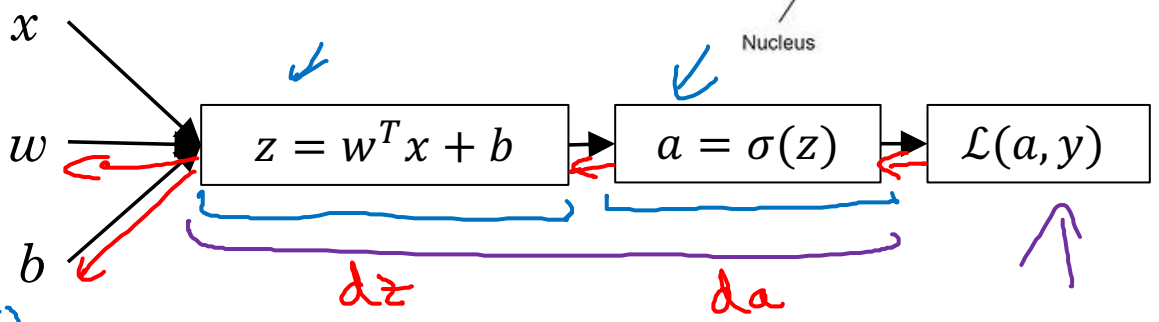
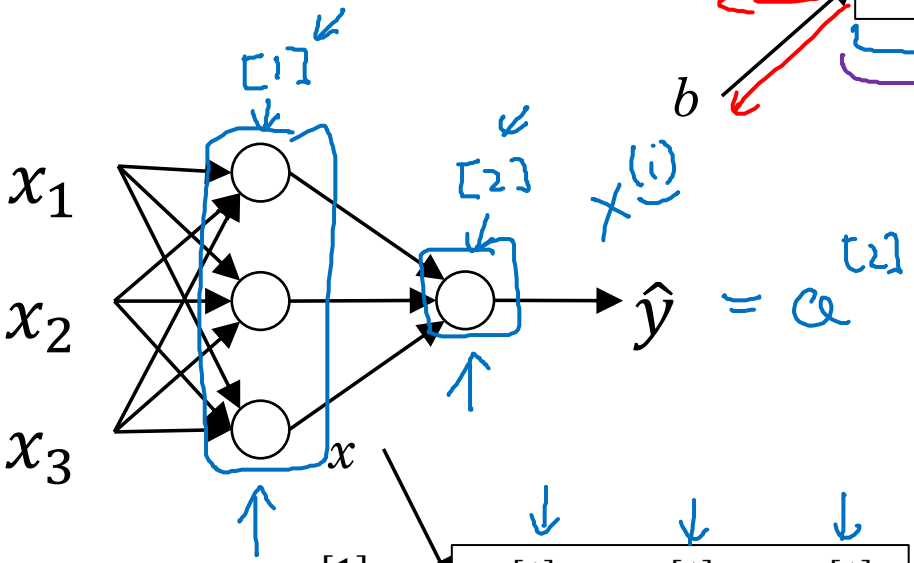
What is a Neural Network?

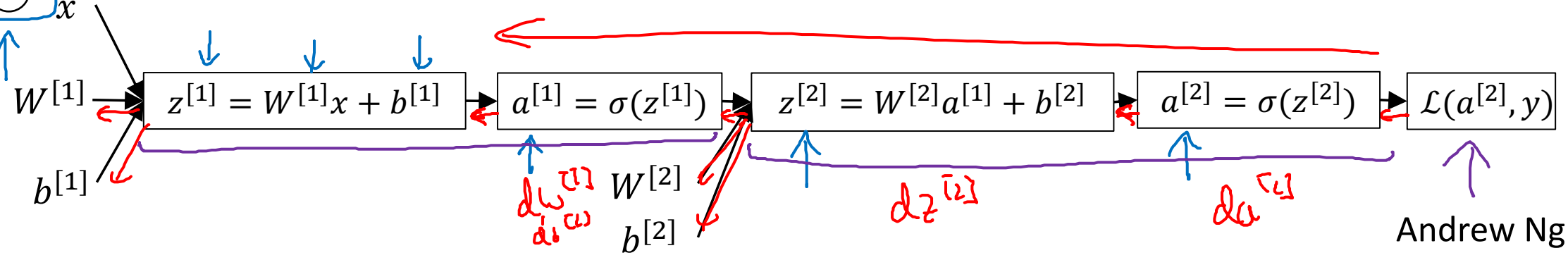






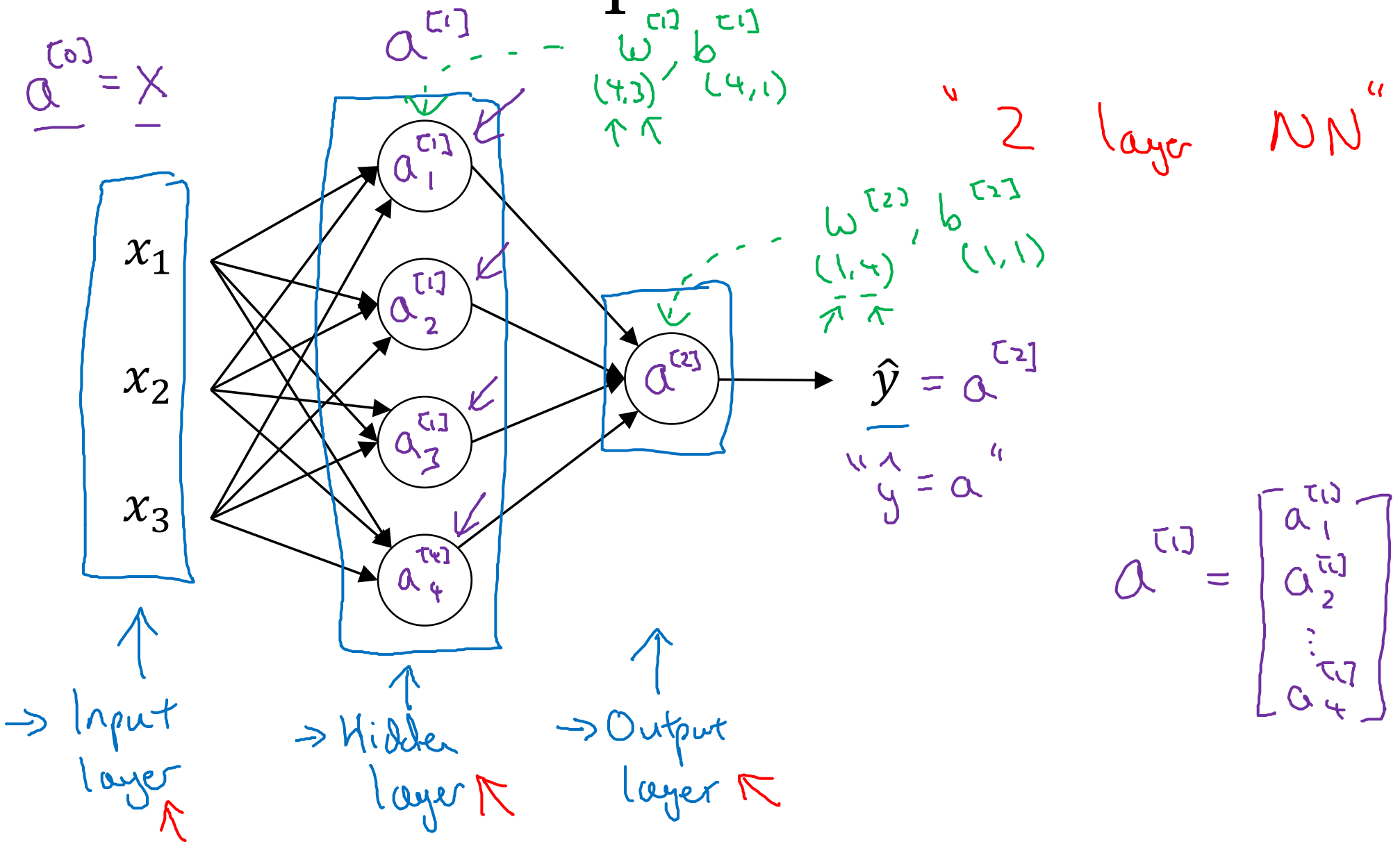






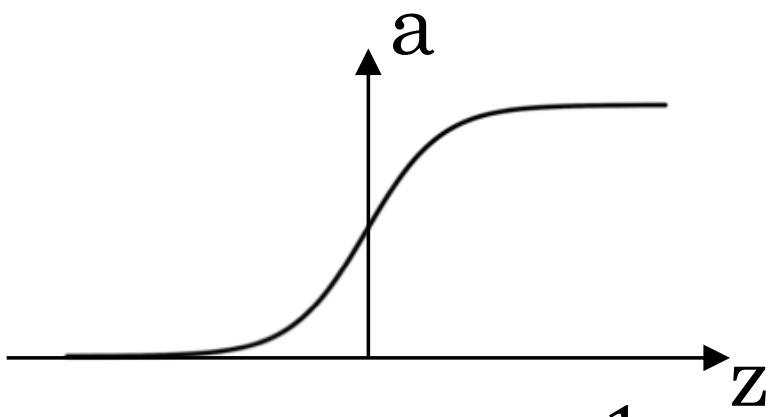
Andrew Ng

Neural Network Representation

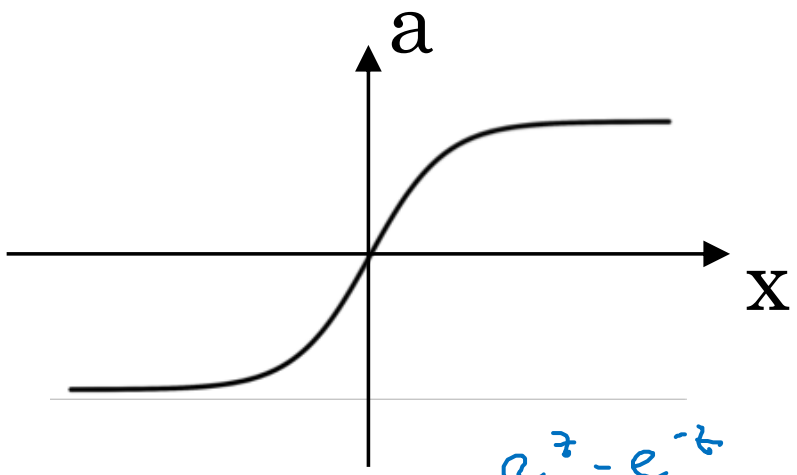


Andrew Ng

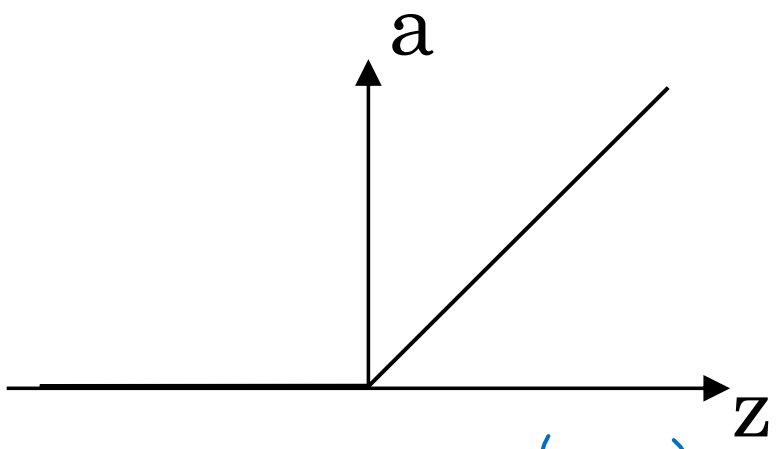
Pros and cons of activation functions



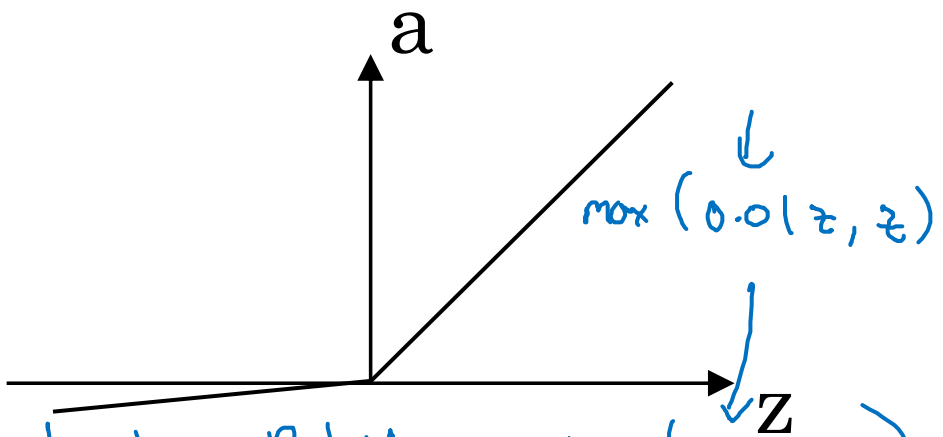
sigmoid: $a = \frac{1}{1+e^{-z}}$



tanh: $a = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$



ReLU $a = \max(0, z)$



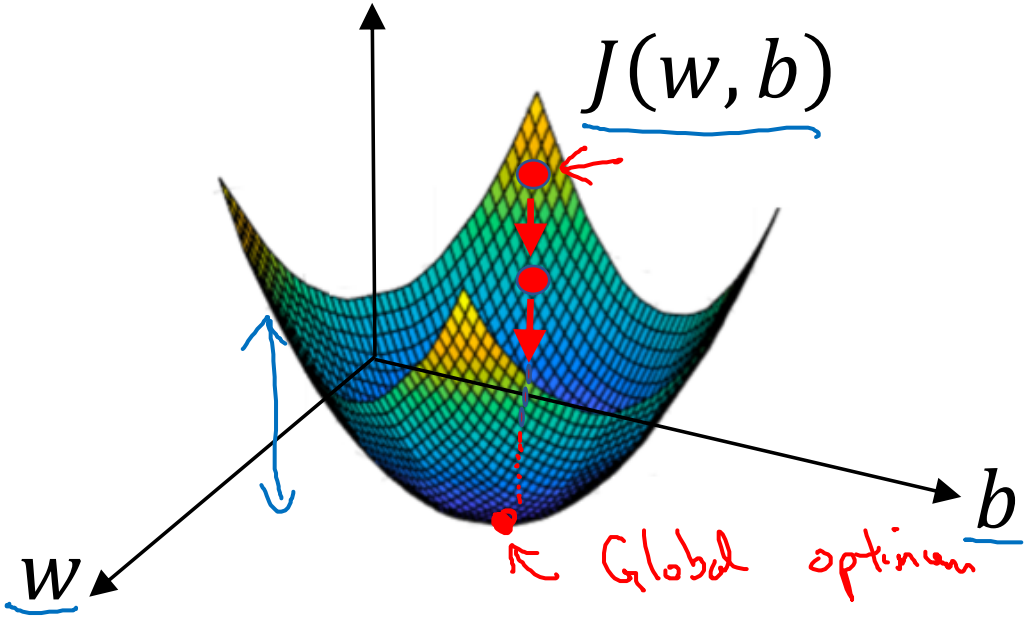
Leaky ReLU $a = \max(0.01z, z)$

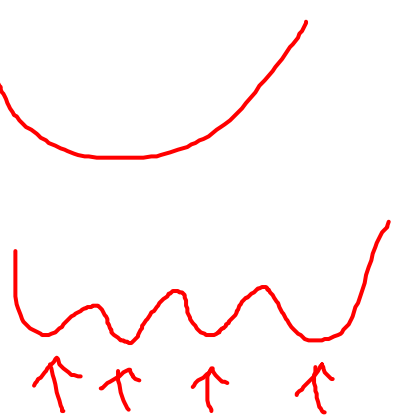
Gradient Descent

Recap: $\hat{y} = \sigma(w^T x + b)$, $\sigma(z) = \frac{1}{1+e^{-z}}$

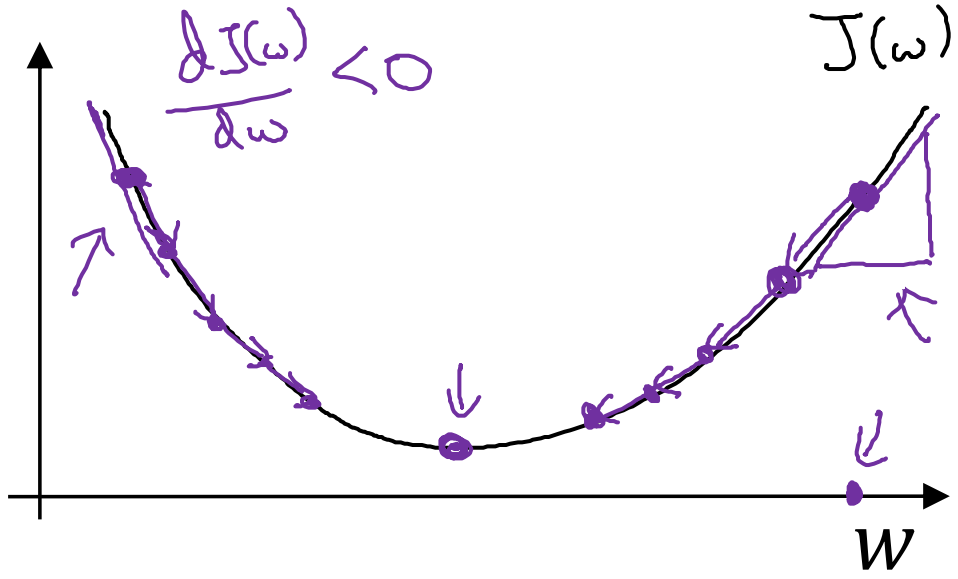
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Want to find w, b that minimize $J(w,b)$





Gradient Descent



Repeat {

$w := w - \alpha \frac{dJ(w)}{dw}$

learning rate

"dw"

$w := w - \alpha dw$

}

$\frac{dJ(w)}{dw} = ?$

$J(w,b)$

$w := w - \alpha \frac{\partial J(w,b)}{\partial w}$

$b := b - \alpha \frac{\partial J(w,b)}{\partial b}$

$\frac{\partial J(w,b)}{\partial w}$ → dw

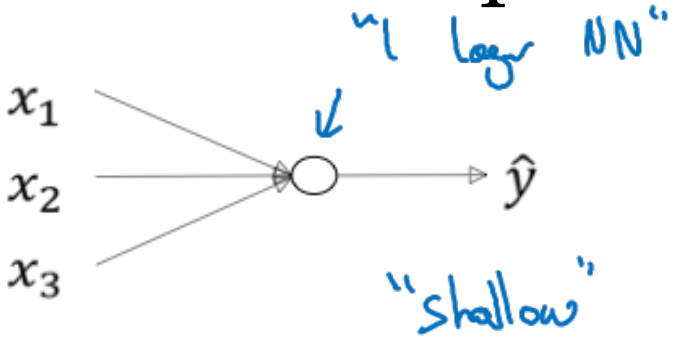
$\frac{\partial J(w,b)}{\partial b}$ → db

∂ "partial derivative"

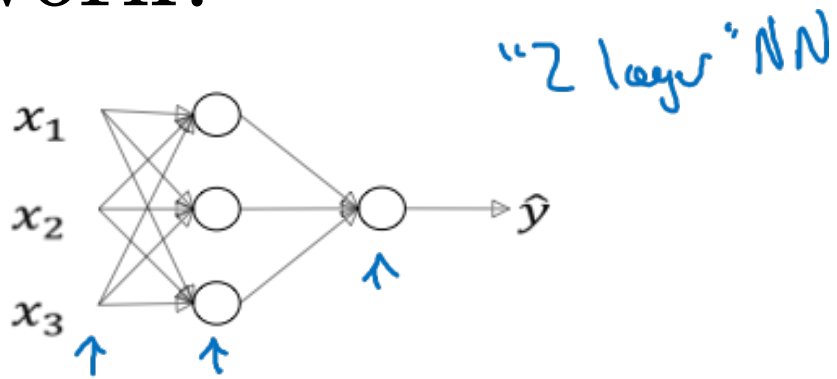
d ← J

Andrew Ng

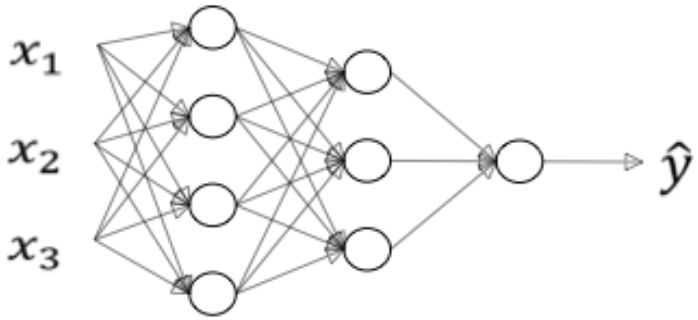
What is a deep neural network?



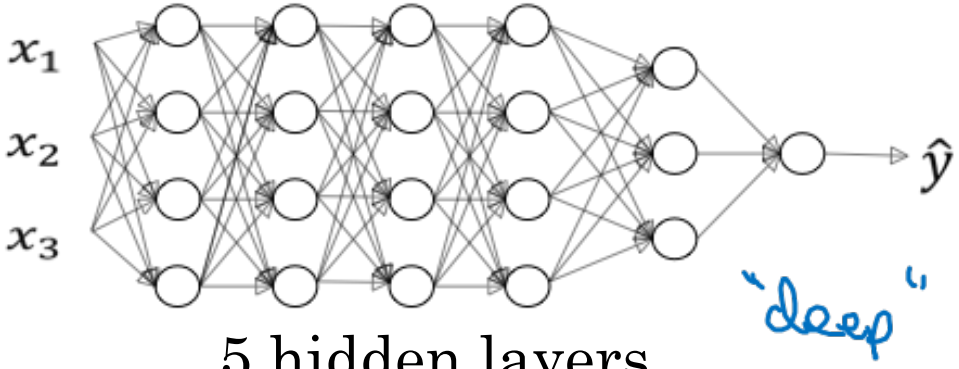
logistic regression



1 hidden layer



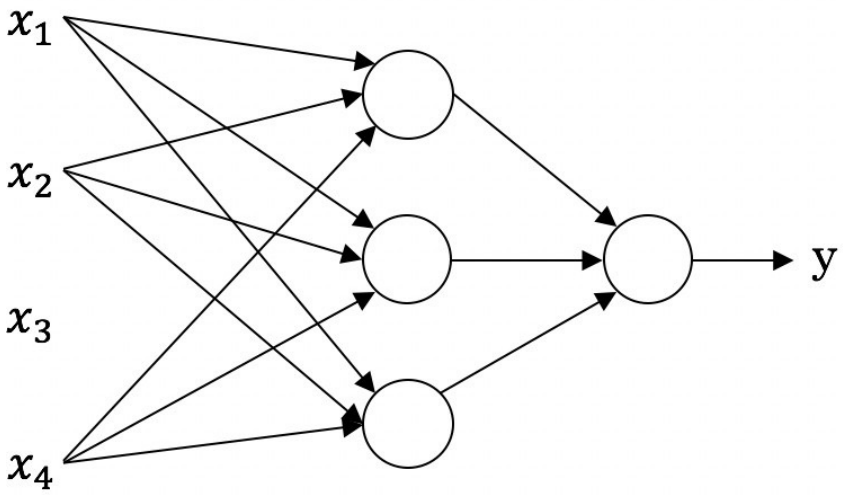
2 hidden layers



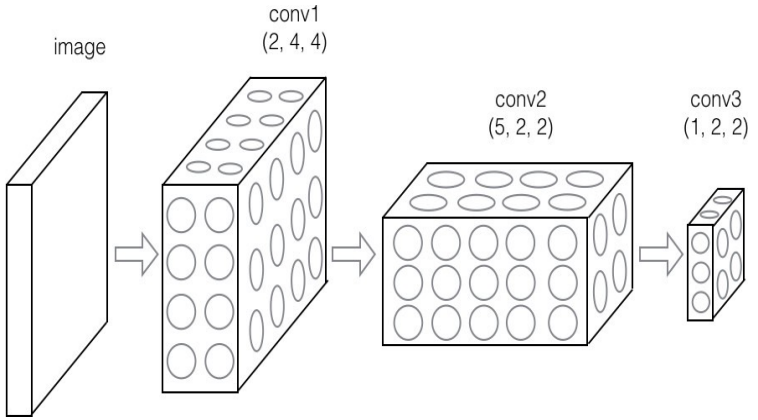
5 hidden layers

Andrew

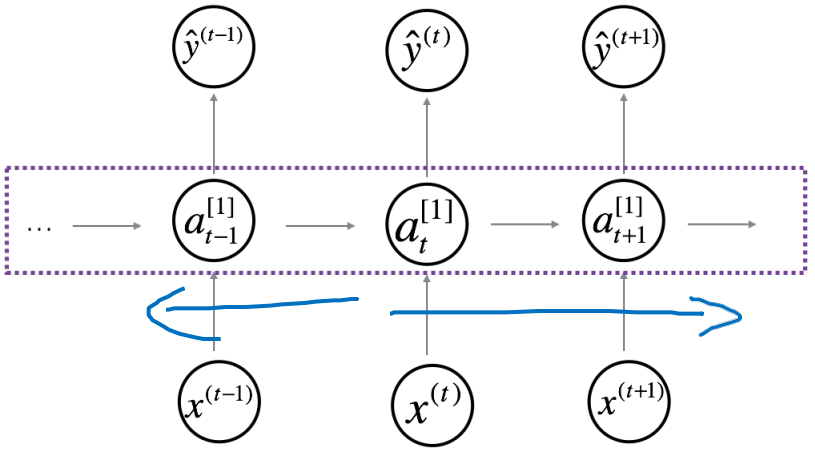
Neural Network examples



Standard NN



Convolutional NN



Recurrent NN

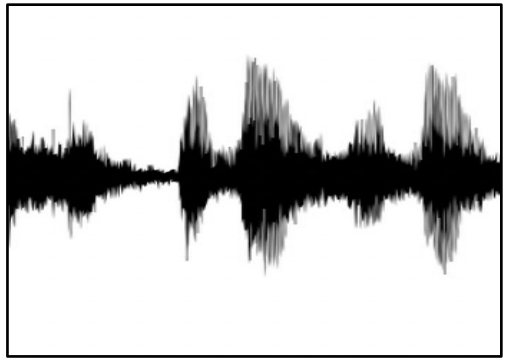
Supervised Learning

Structured Data

Size	#bedrooms	⋯	Price (1000$s)
2104	3		400
1600	3		330
2400	3		369
⋯	⋯		⋯
3000	4		540

User Age	Ad Id	⋯	Click
41	93242		1
80	93287		0
18	87312		1
⋯	⋯		⋯
27	71244		1

Unstructured Data



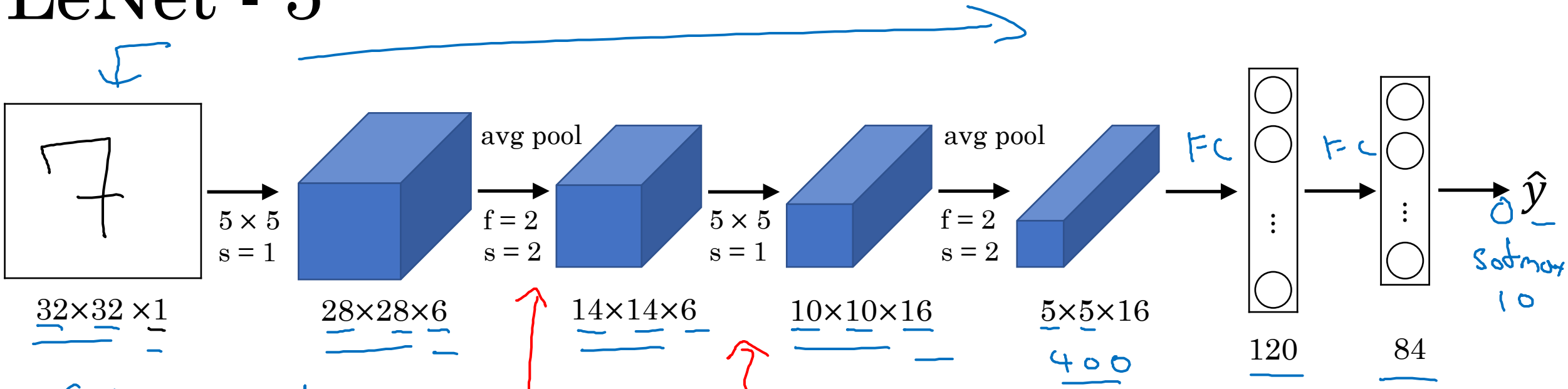
Audio

Image

Four scores and seven years ago…

Text

LeNet - 5



60k parameters.

$n_H, n_w \downarrow$ $n_c \uparrow$

Non-linearity after pooling

$n_{H} \times n_{w} \times n_c$ $f \times f \times n_c$

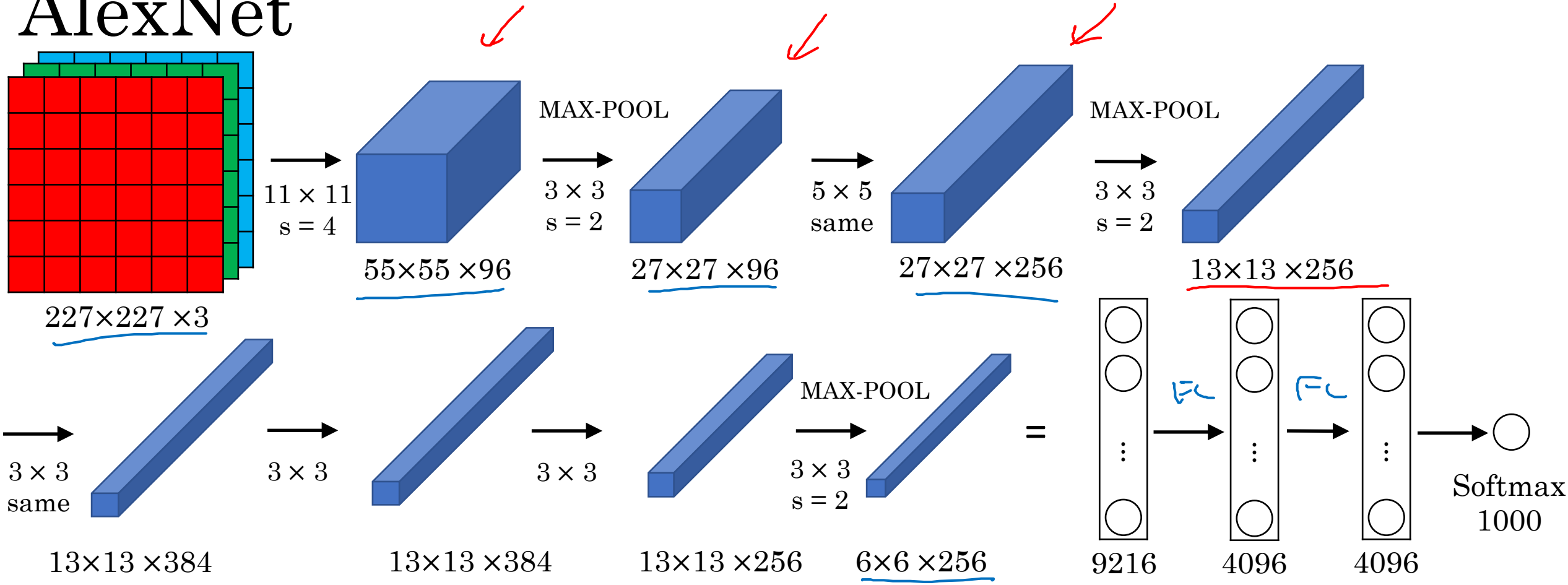
conv pool conv pool fc fc output

Activation: sigmoid/tanh ReLU

II III

[LeCun et al., 1998. Gradient-based learning applied to document recognition]

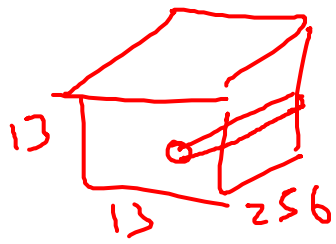
AlexNet



- Similar to LeNet, but much bigger.
- ReLU
- Multiple GPUs.
- Local Response Normalization (LRN)

9216

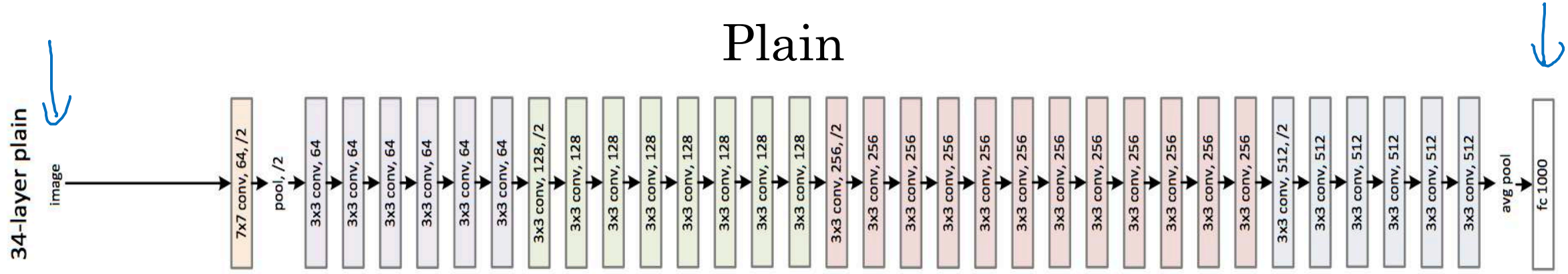
60M parameters



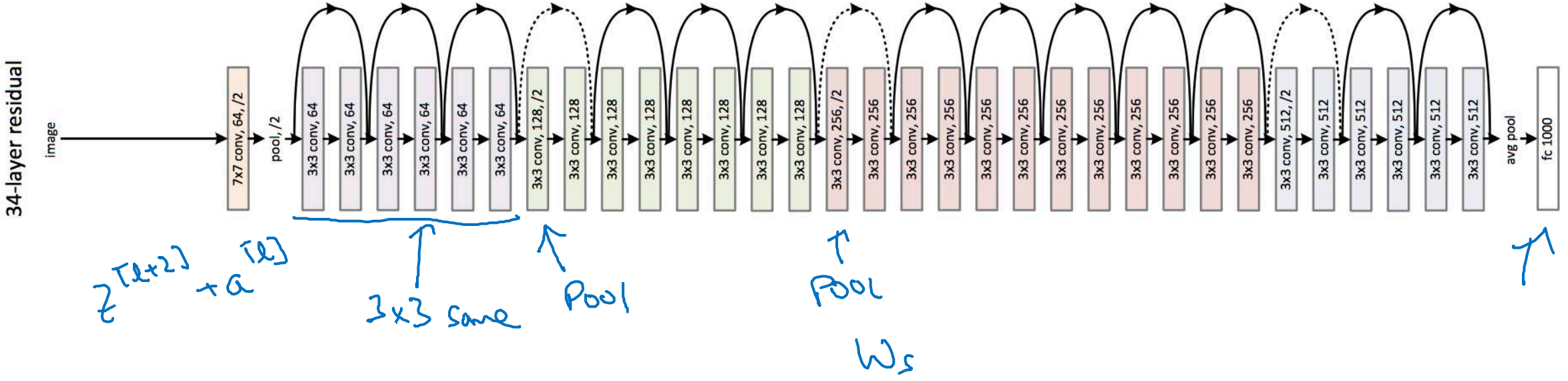
[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

ResNet

Plain



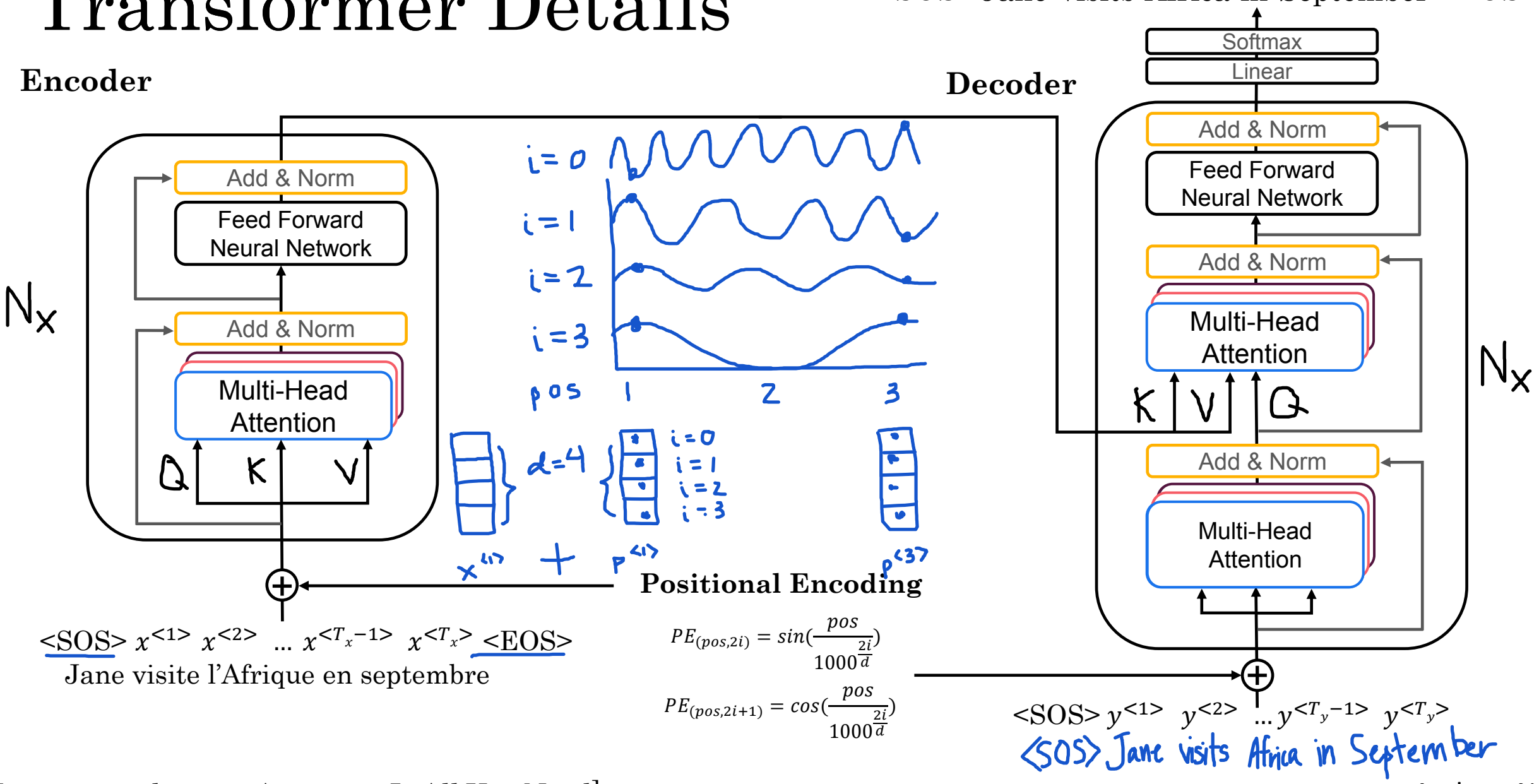
ResNet



[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

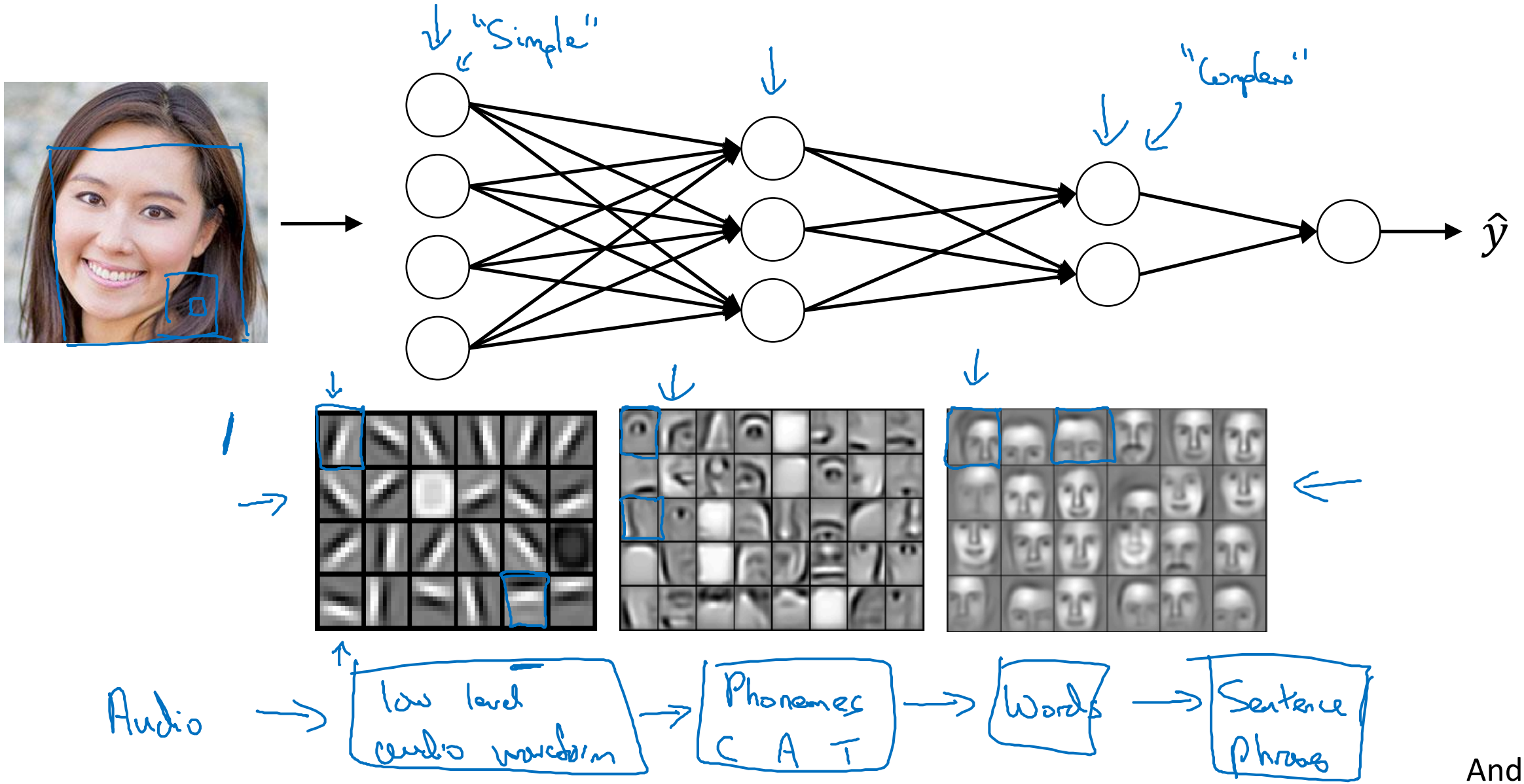
Transformer Details



[Vaswani et al. 2017, Attention Is All You Need]

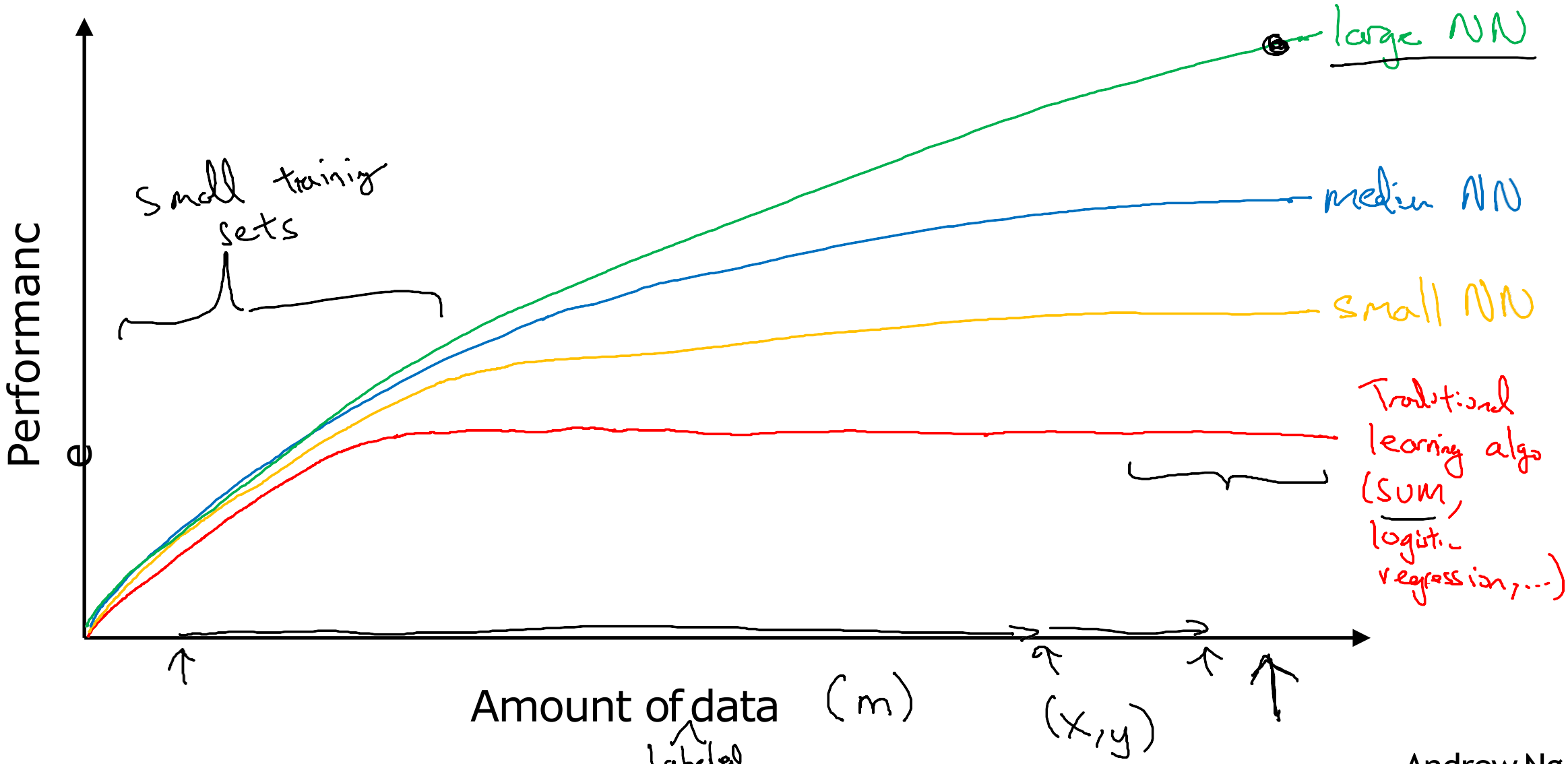
Andrew Ng

Intuition about deep representation



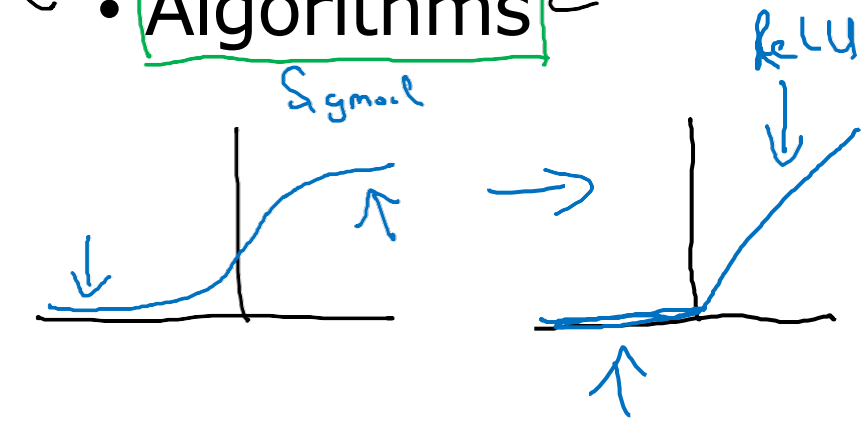
Andrew Ng

Scale drives deep learning progress

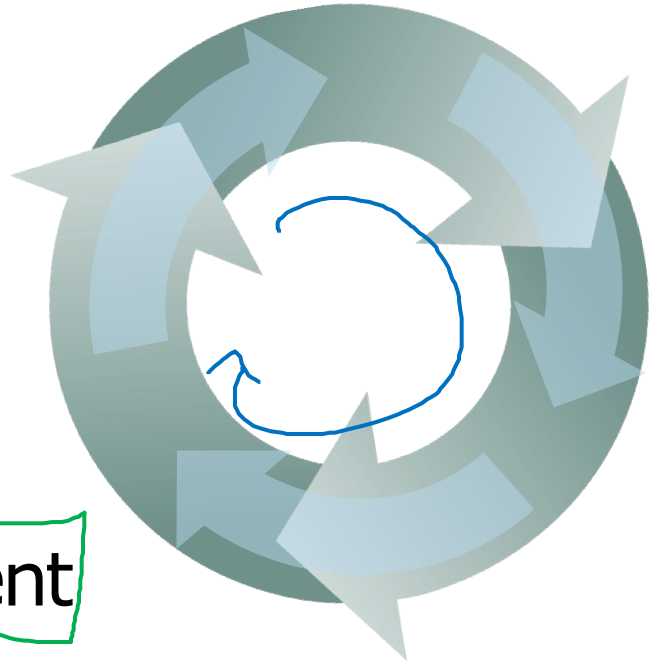


Scale drives deep learning progress

- Data
- Computation
- Algorithms



Idea



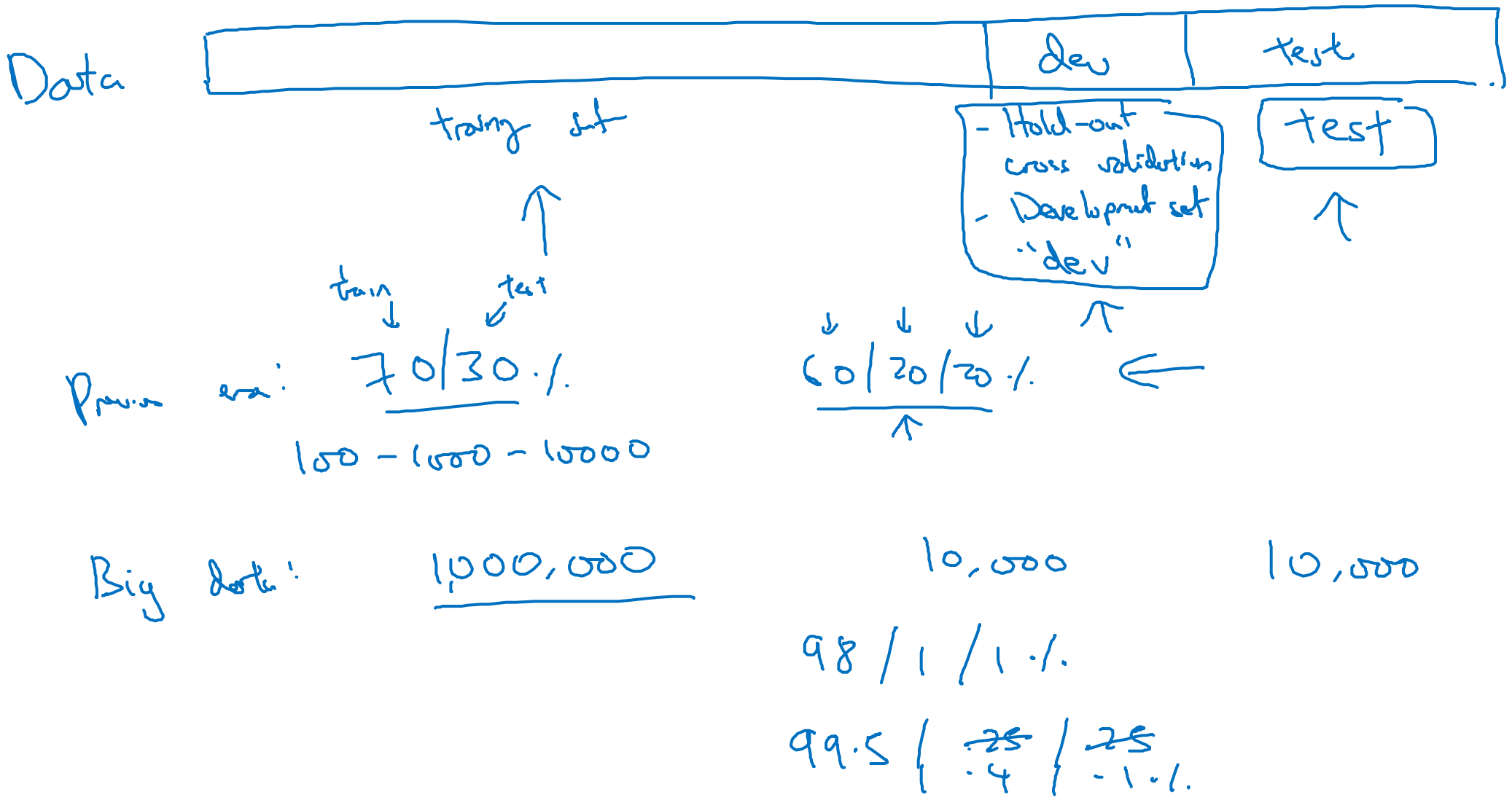
Experiment

Code

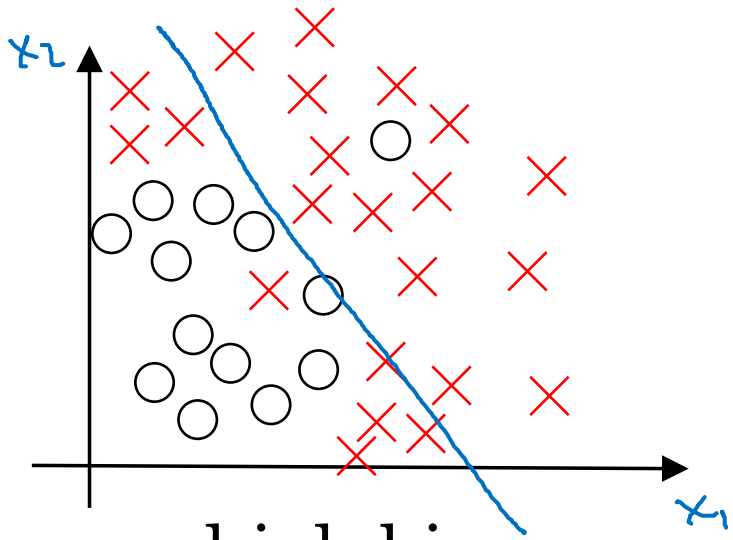
10 min
1 day
1 month

Andrew Ng

Train/dev/test sets

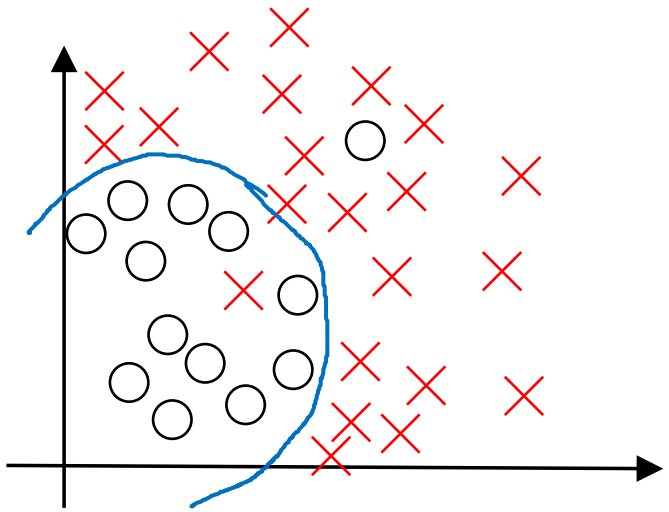


Bias and Variance

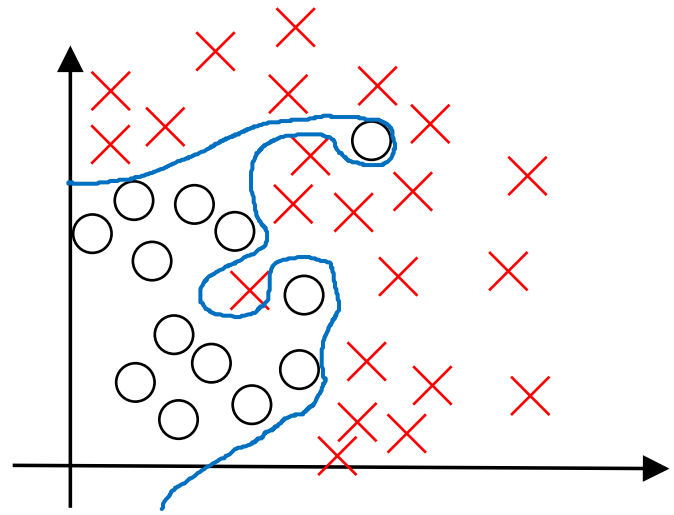


high bias

Underfitting



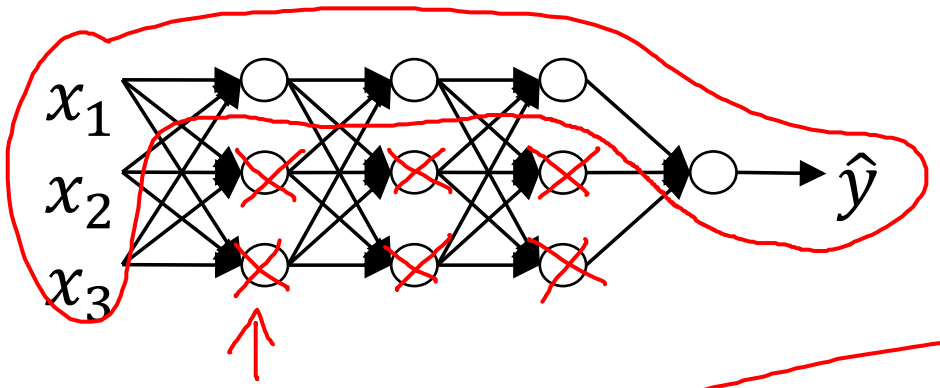
→ "just right"



high variance

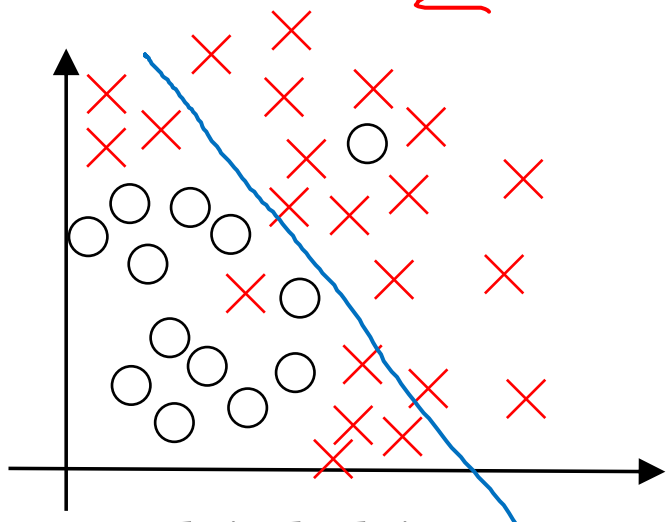
Overfitting

How does regularization prevent overfitting?

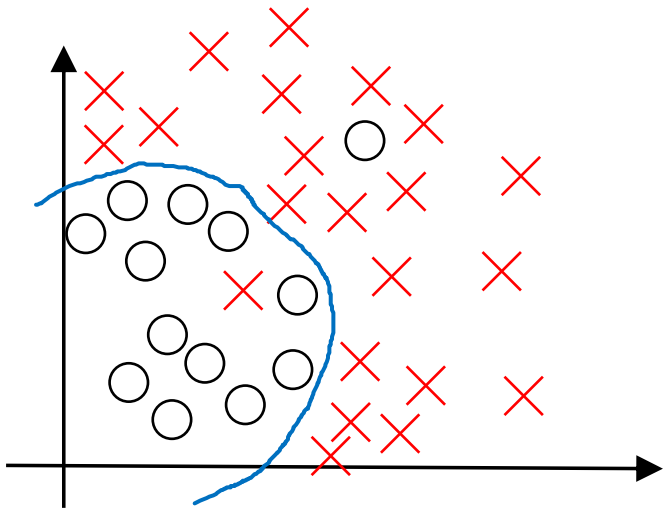


$$J(w^{[l]}, b^{[l]}) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L} \|w^{[l]}\|_F^2$$

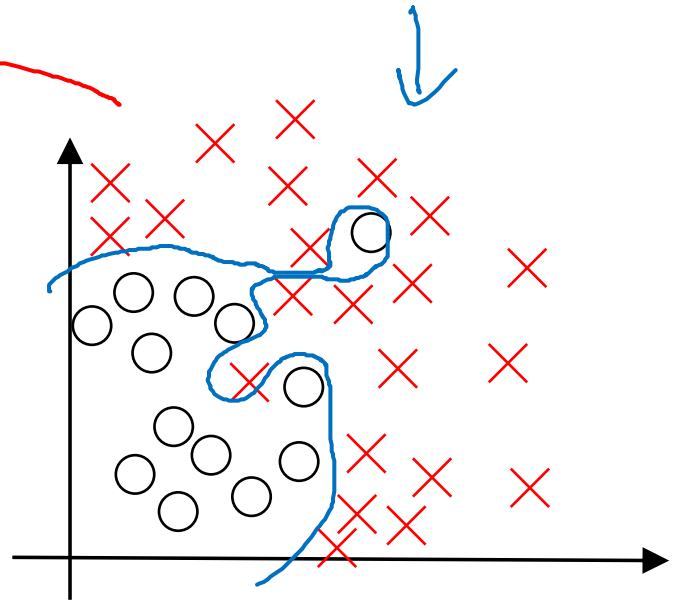
$w^{[l]} \approx 0$



high bias



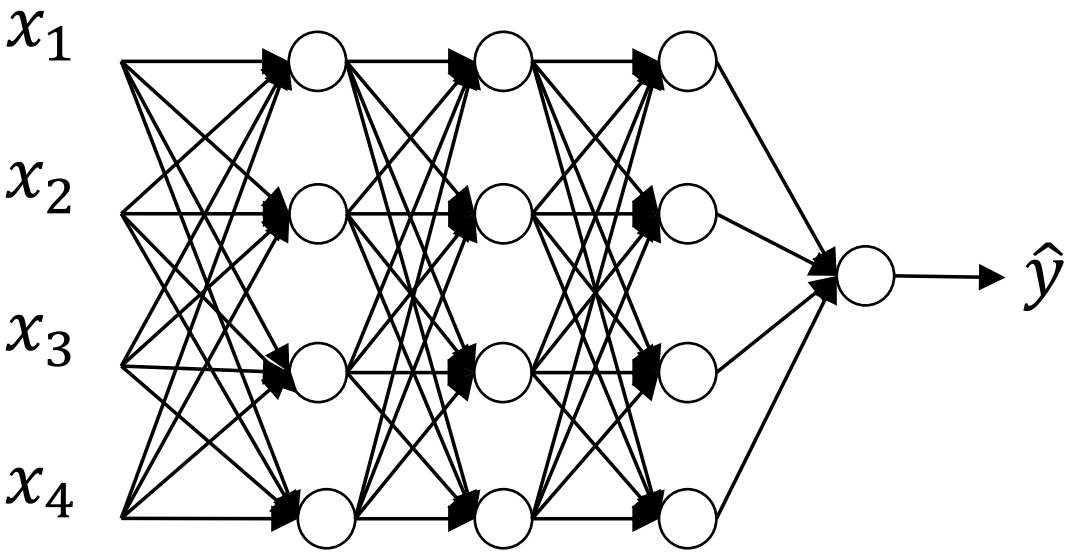
"just right"

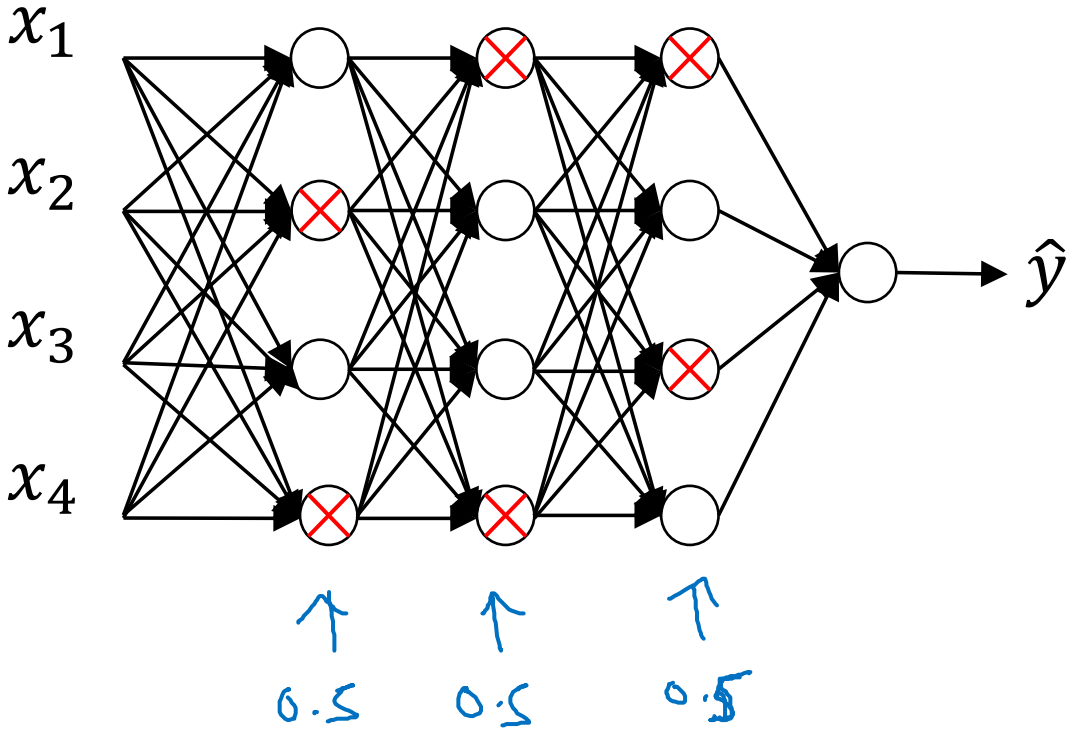


high variance

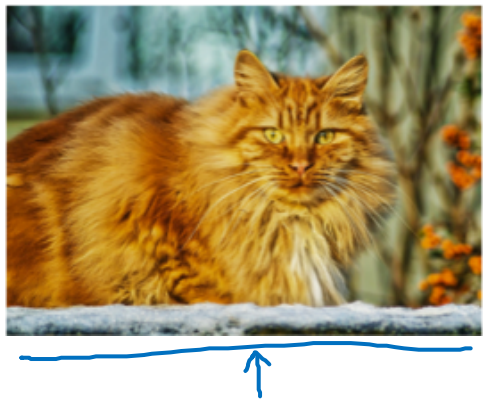
Andrew Ng

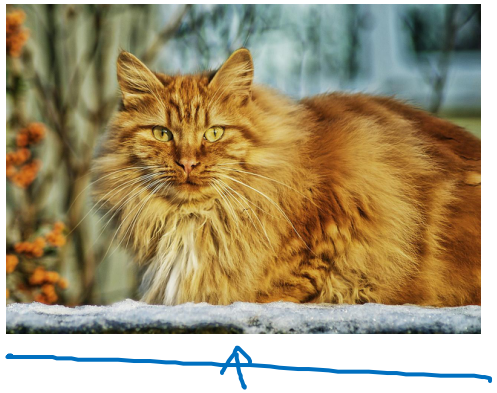
Dropout regularization



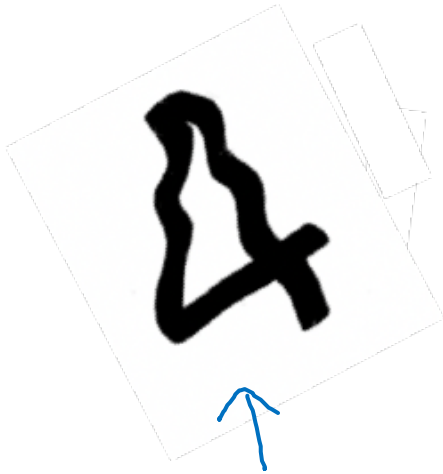


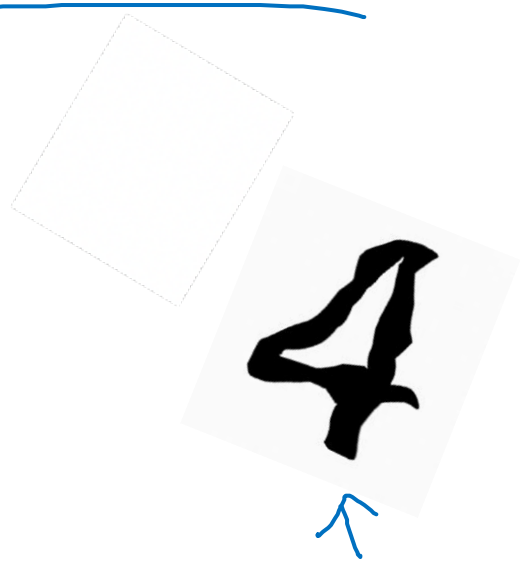
Data augmentation





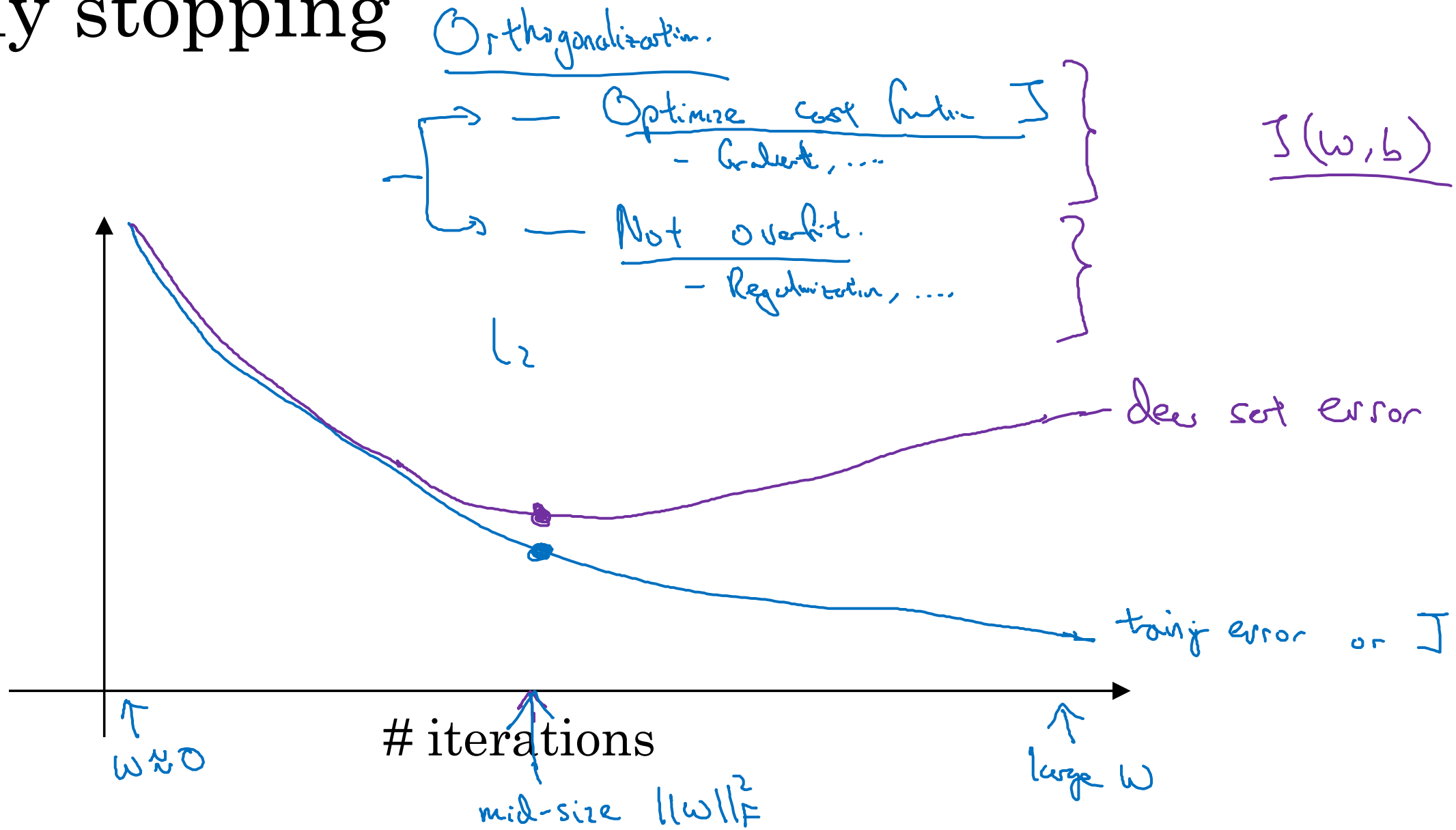
4





Andrew Ng

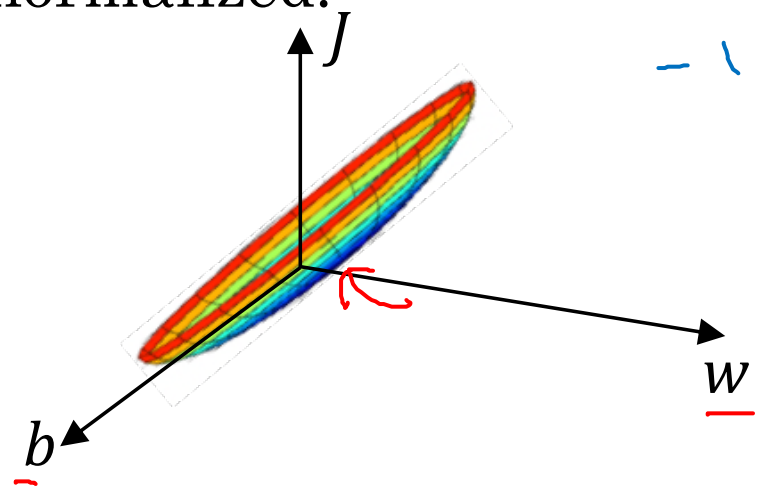
Early stopping



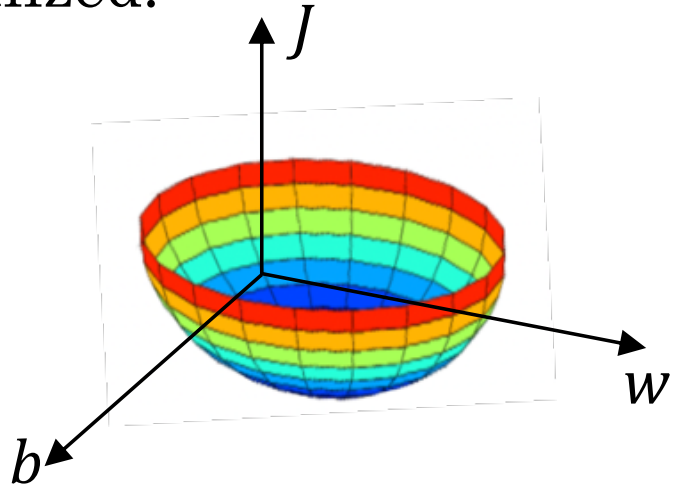
Why normalize inputs?

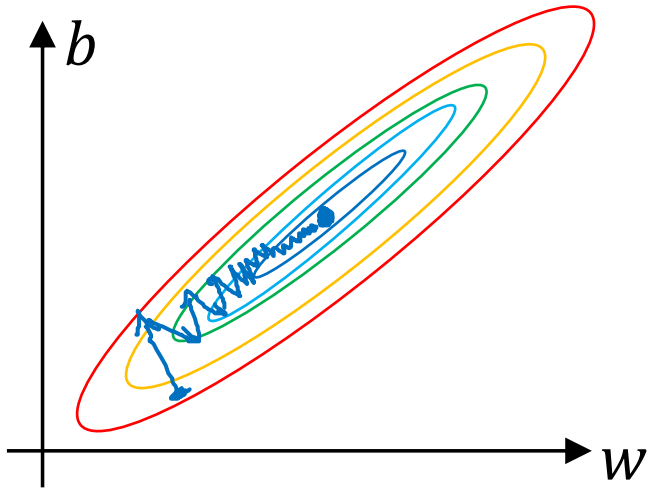
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

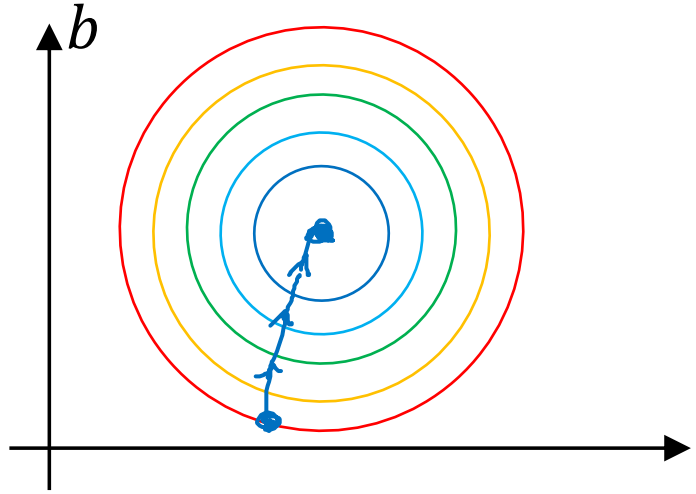
Unnormalized:



Normalized:

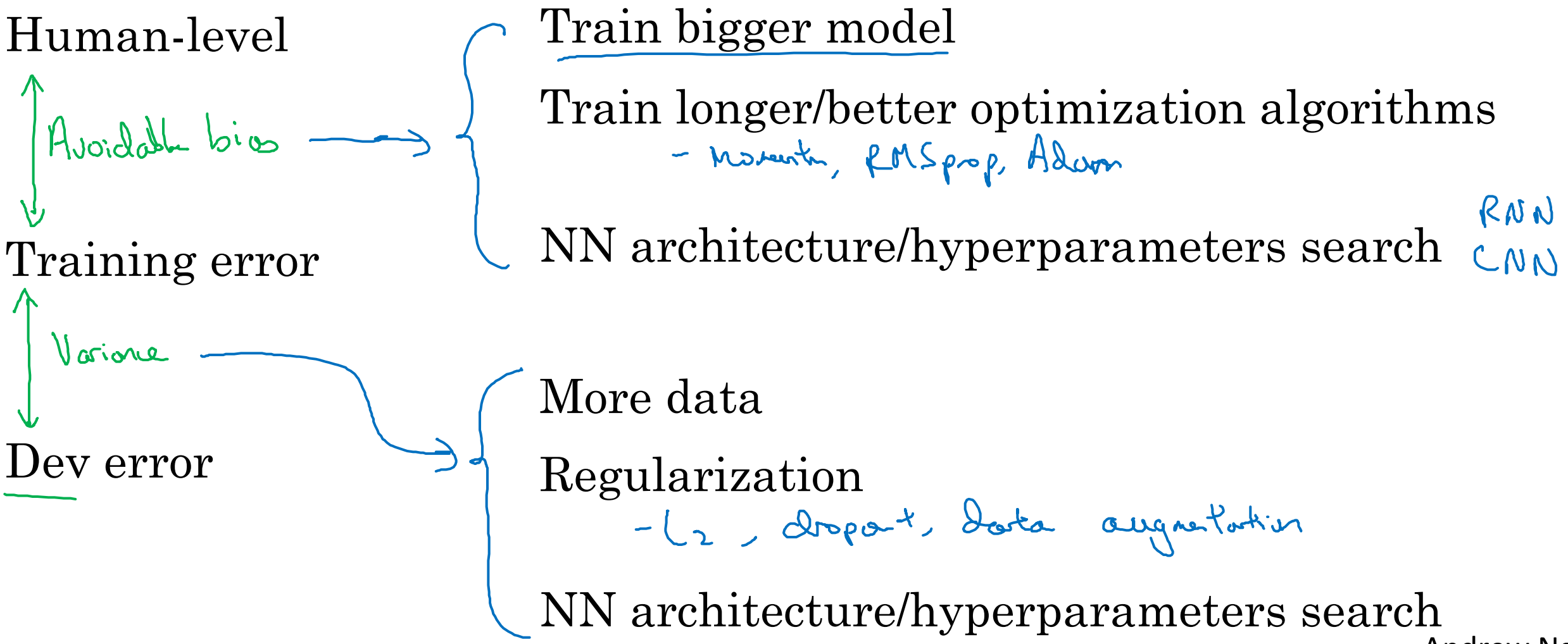






Andrew Ng

Reducing (avoidable) bias and variance



Copyright Notice