Deep Learning

Big Data & Machine Learning Bootcamp - Keep Coding

© All rights reserved. www.keepcoding.io

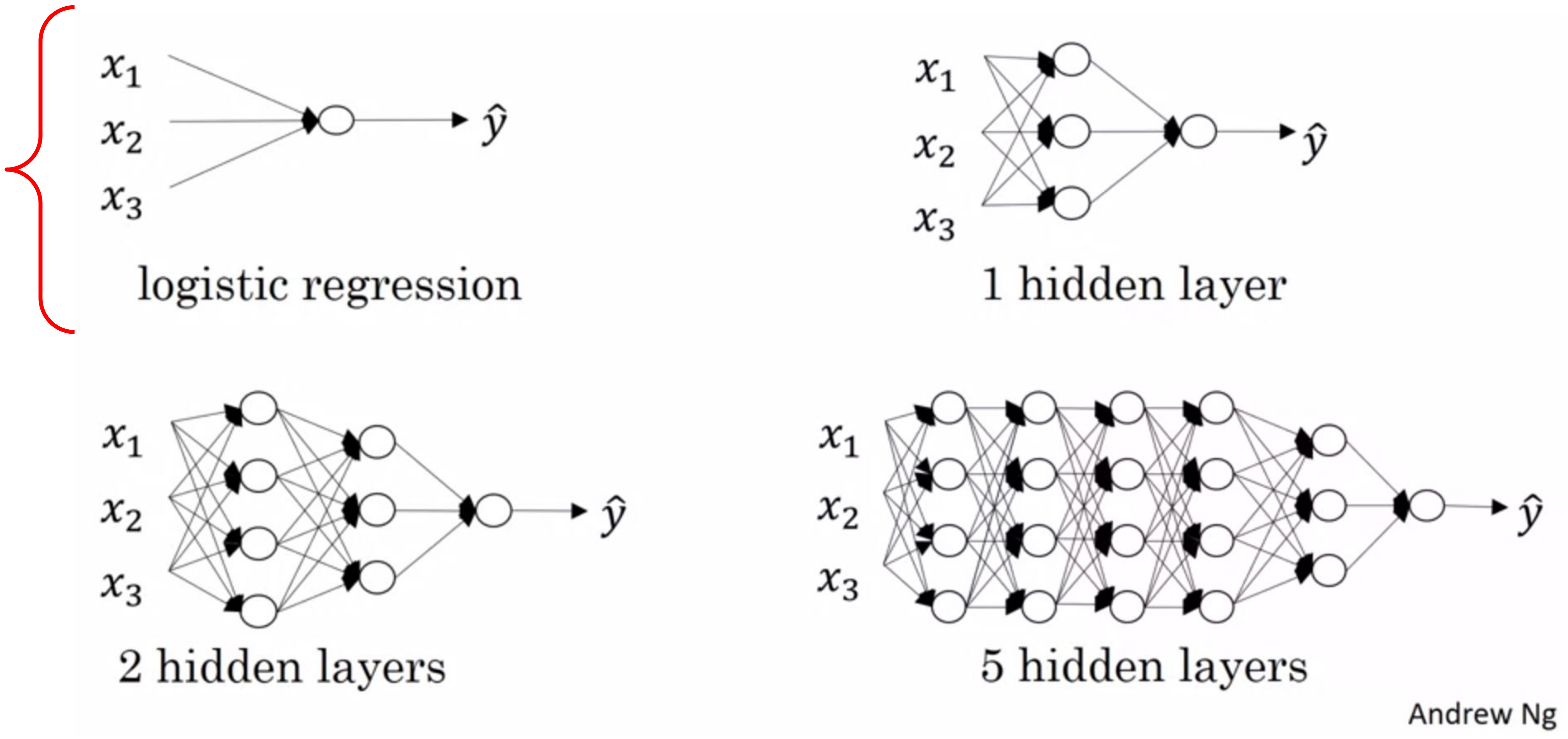
Outline

1. Deep Neural Network
2. Why deep representations?
3. Regularization
4. Normalizing Inputs
5. Vanishing/Exploding gradients

© All rights reserved. www.keepcoding.io

Deep Neural Network

“Shallow network”



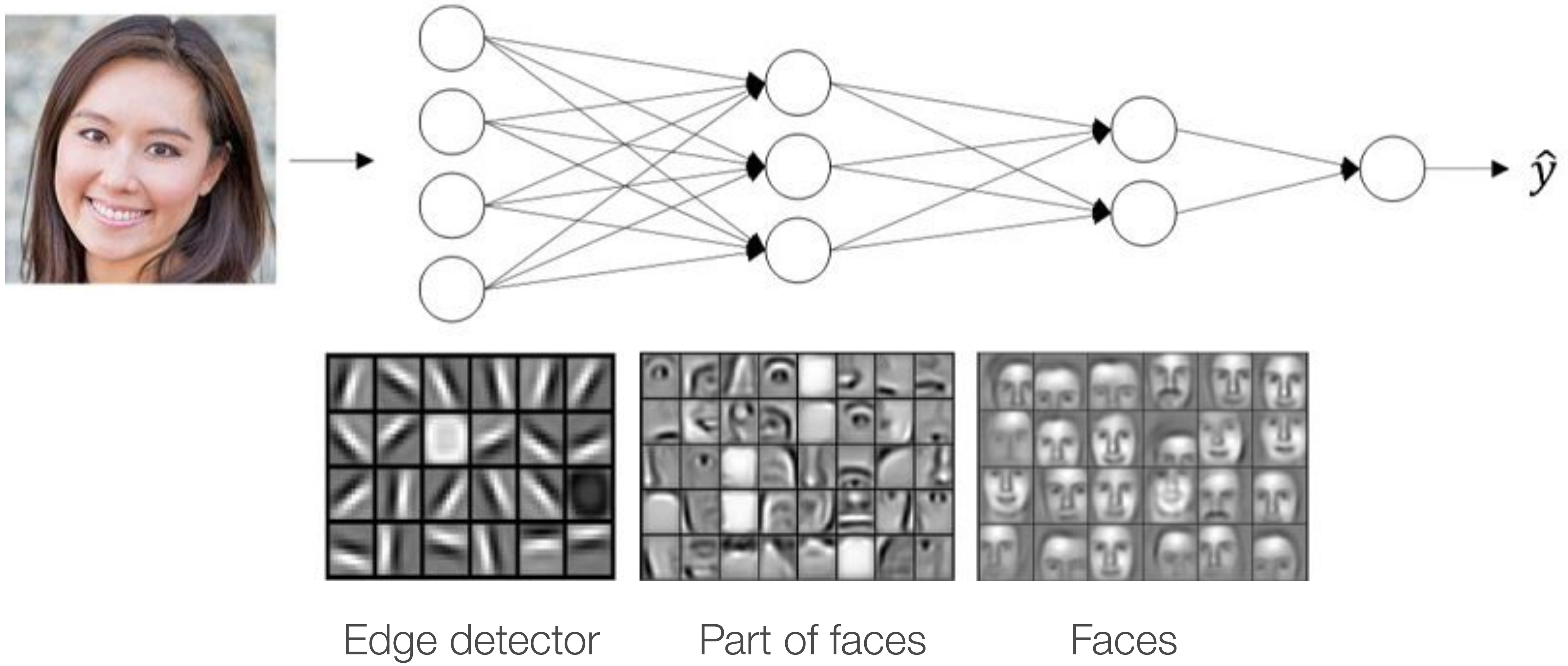
We don’t count the input and the output layers. We only count the hidden layers!

Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Why deep representations?

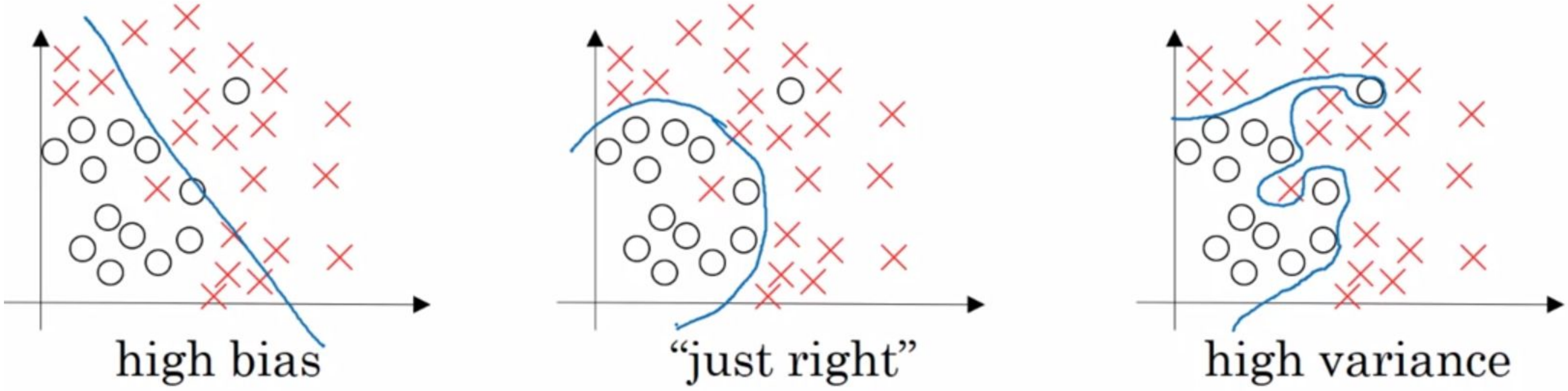
The first layers detect low level features while deep layers detect more complex features



Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Bias/variance trade-off



Underfitting ***Overfitting***

Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Train - Development and Test sets

Train set error	1%	15%	15%	0.5%
Dev set error	11%	16%	30%	1%
	High variance (Overfitting)	High bias (Underfitting)	High bias & high variance	Low bias and low variance

This is assuming that the human error or the optimal error is 0%

We'll discuss this later. But this doesn't occur in real life. ***The optimal error is around 15%***

Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Regularization

L2 regularization: Add a term to the loss function that is function of the parameters in he neural network

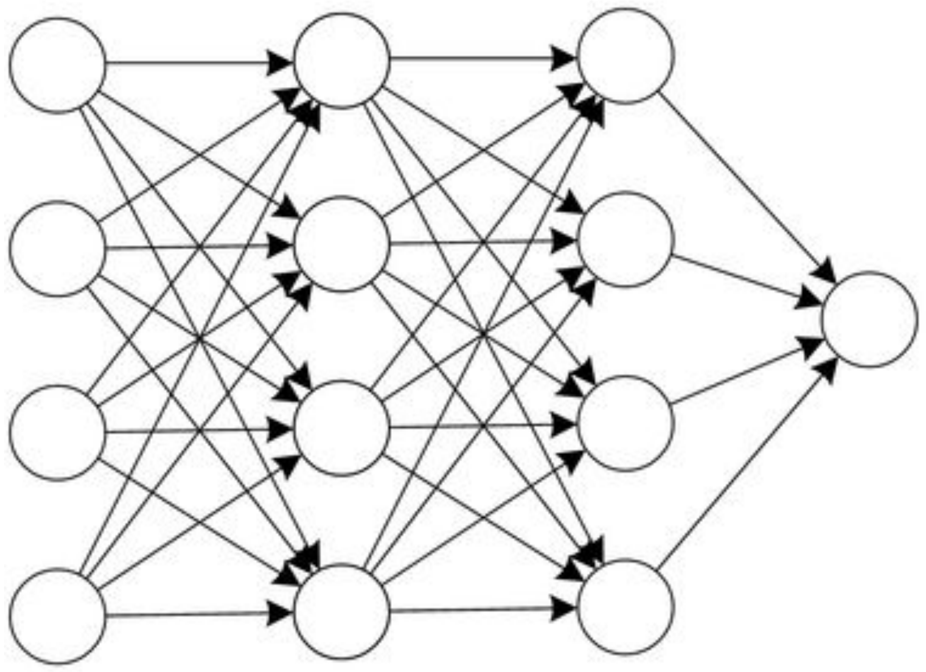
L2 regularization is also called **weight decay.**

Sources:
- Coursera

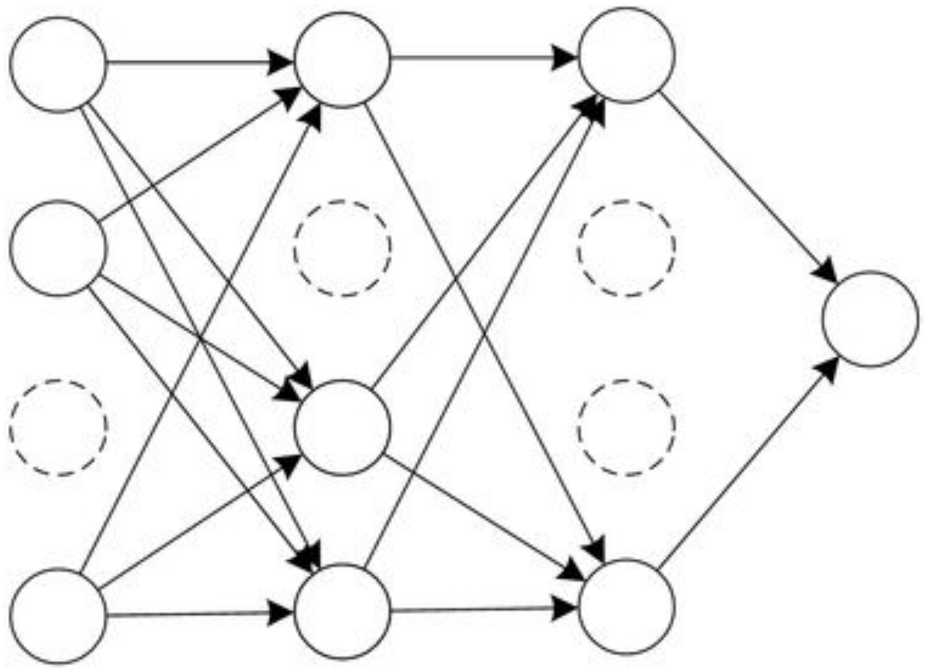
© All rights reserved. www.keepcoding.io

Regularization

Dropout is also another regularization technique! How does it work?



(a) Standard Neural Network



(b) Network after Dropout

Essentially you define a ***probability of keeping a neuron.***

A common way of implementing dropout is using the “inverted dropout”

© All rights reserved. www.keepcoding.io

Sources:
- Coursera
- Amine ben khalifa et al. Multiple Instance Fuzzy Inference Neural Networks

Regularization

There are other regularization techniques such as:

- **Data augmentation** (mirroring, horizontal and vertical rotation, zooming, etc)
- **Early stopping:** You stop the training process when the dev set error gets bigger when compared to the training set error.

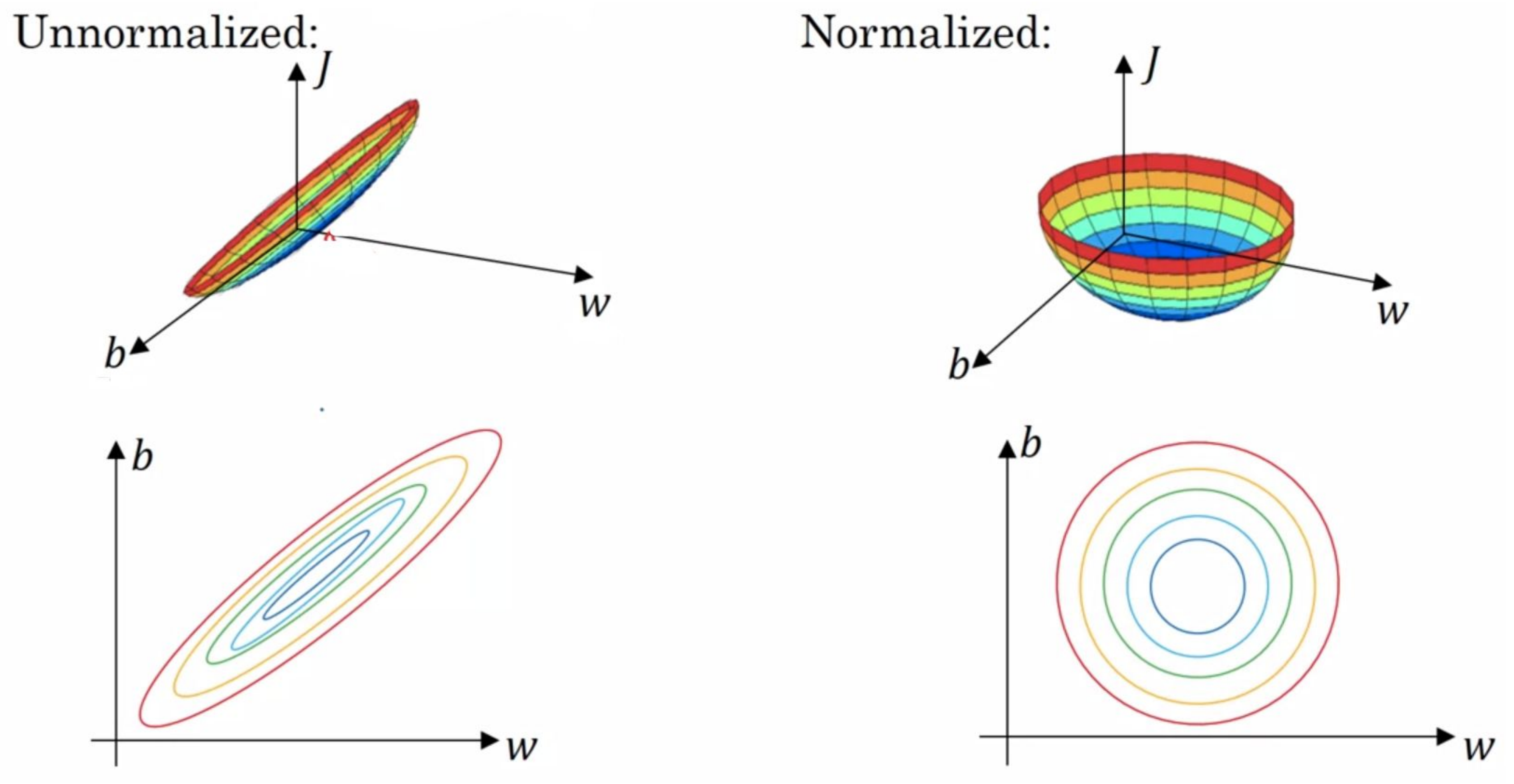
  Early stopping is **not** recommended as it breaks orthogonalization. This means, orthogonalization doesn't allow you to work on optimizing the cost function and avoiding overfitting independently.

Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Normalizing Inputs

The main reason for normalizing the inputs is that the cost function is more rounded and easy to optimize!



The main idea is to keep the similar ranges among the input variables

Sources:
- Coursera

© All rights reserved. www.keepcoding.io

Vanishing/Exploding gradients

When training very deep neural networks, **slopes or the derivatives can get very big (exploding) or very very small (vanishing)**

To remember:

- Weight values that are smaller than 1 make the updates to decrease when doing multiplication ***(vanishing gradients)***
- The contrary happens when having derivative values bigger than 1, updates increase ***(exploding gradients)***

A proper weight initialization method may help to reduce this problem!

Sources:
- Coursera

© All rights reserved. www.keepcoding.io