ARTIFICIAL NEURAL NETWORKS II

Prof. Alexander Huth
3.31.2020

COURSE ADMIN / ZOOM

* Welcome to the new zoom era
* If you’re watching this, you’ve found the lecture links posted on Canvas
* This (and future) lectures will be **recorded** and shared with the class via Canvas
  * If you don’t want your face recorded, please disable your video!
  * Please don’t share the recordings outside of class!
* And please don’t share the zoom link outside of class!

COURSE ADMIN / ZOOM

* Current plan for zoom etiquette:
  * Please **keep yourselves muted** except when you are actively speaking
  * If you want to ask a question, **use the “Raise hand” button** and I’ll call on you
  * If I miss your raised hand (apologies in advance), toggle it a few times, or just un-mute yourself and ask
  * I probably won’t be able to keep track of chat messages while lecturing & screen-sharing, so **please avoid using the chat feature**

COURSE ADMIN / ZOOM

* Office hours will also happen on zoom, Tuesdays & Wednesdays, 10:30am-12:00pm

* I’ll start a zoom meeting & post the link on canvas around when my office hours start

COURSE ADMIN / HW

* Homework 1 is not graded yet, but will hopefully be returned to you in ~1 week

* Homework 2 covers linearized model comparison and variance partitioning. It is posted **today** & will be due in 2 weeks (**April 14**)

COURSE ADMIN / PROJECT

* Goal is to apply or explore something / anything we've talked about in this class
  * could be using real data (e.g. fit some kind of model to a neural dataset)
  * could be theory/methods (e.g. find a better way to do something)

COURSE ADMIN / PROJECT

* Since there are so many people in this class, **I will require you to work on your projects in groups of 2-4**

* If you can’t find a group, please contact me or come to office hours to discuss

COURSE ADMIN / PROJECT

* Proposal due **next Thursday** (**April 9**):
  * ~1-2 paragraphs describing what you plan to do & who is in your group. Email to huth@cs.utexas.edu before class
* Writeup (3-4 pages explaining background & what you did) due **May 5**
* In-class presentations (5-10 minutes) **May 5 & 7**

COURSE ADMIN / PROJECT

* Presentations should be live (zoom) and include slides describing the background, your method/approach, and results

* **It is essential** that every person in your group participates in both the project work and the presentation

COURSE ADMIN / PROJECT

* Sources of neural data (among many more):
  * CRCNS: https://crcns.org/data-sets
  * Allen Inst.: http://www.brain-map.org
  * Study Forrest: http://studyforrest.org/

COURSE ADMIN / ?

* Any further questions about any of these administrative things?

RECAP:
NONLINEAR METHODS

* **Volterra series**

* **Kernel regression** *(samples, not features!)*

* **Artificial neural network** *(1 hidden layer)*

ARTIFICIAL NEURAL NETWORKS

* We previously talked about **Perceptrons**, simple 1-layer networks with a step-function output nonlinearity

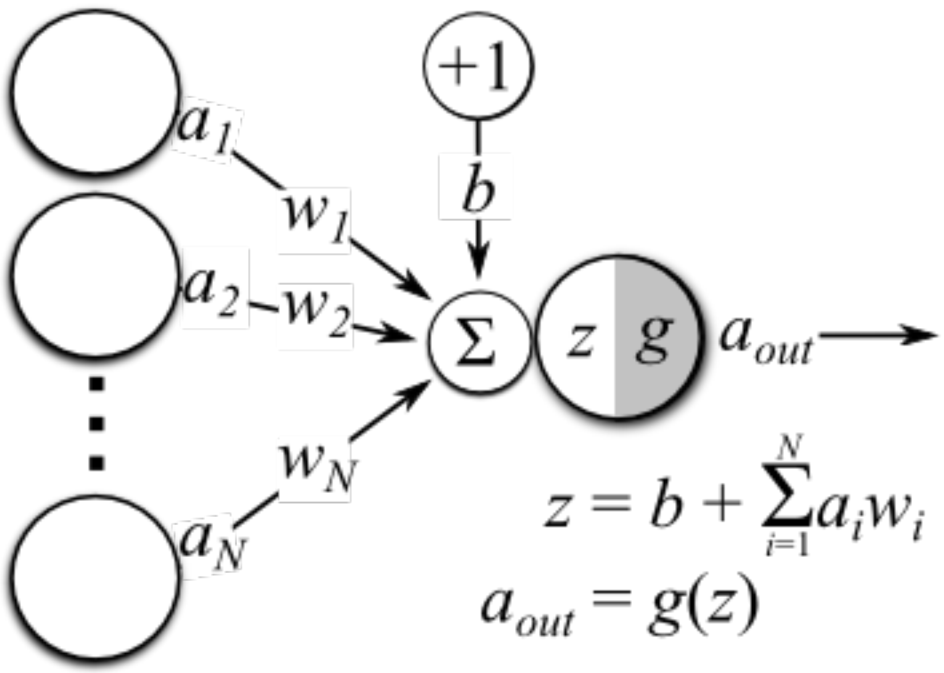
* We looked at the Perceptron learning rule, which can be used to iteratively learn weights

ARTIFICIAL NEURAL NETWORKS

* Generic 1-layer neural "network"

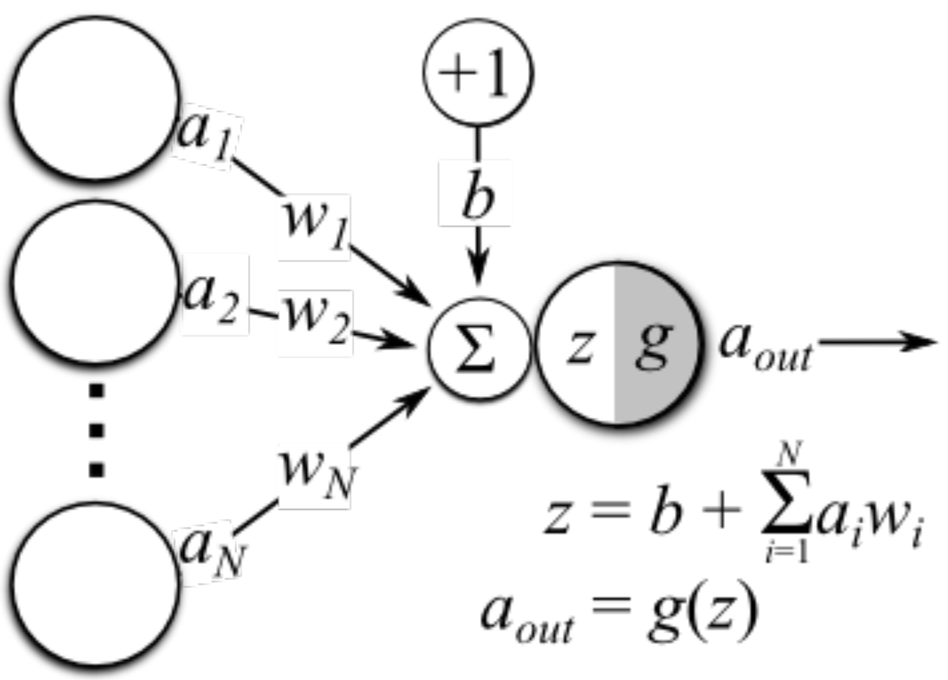
* g(z) is the output nonlinearity

* Update rule? Just differentiate the loss function! (As long as g(z) is differentiable)

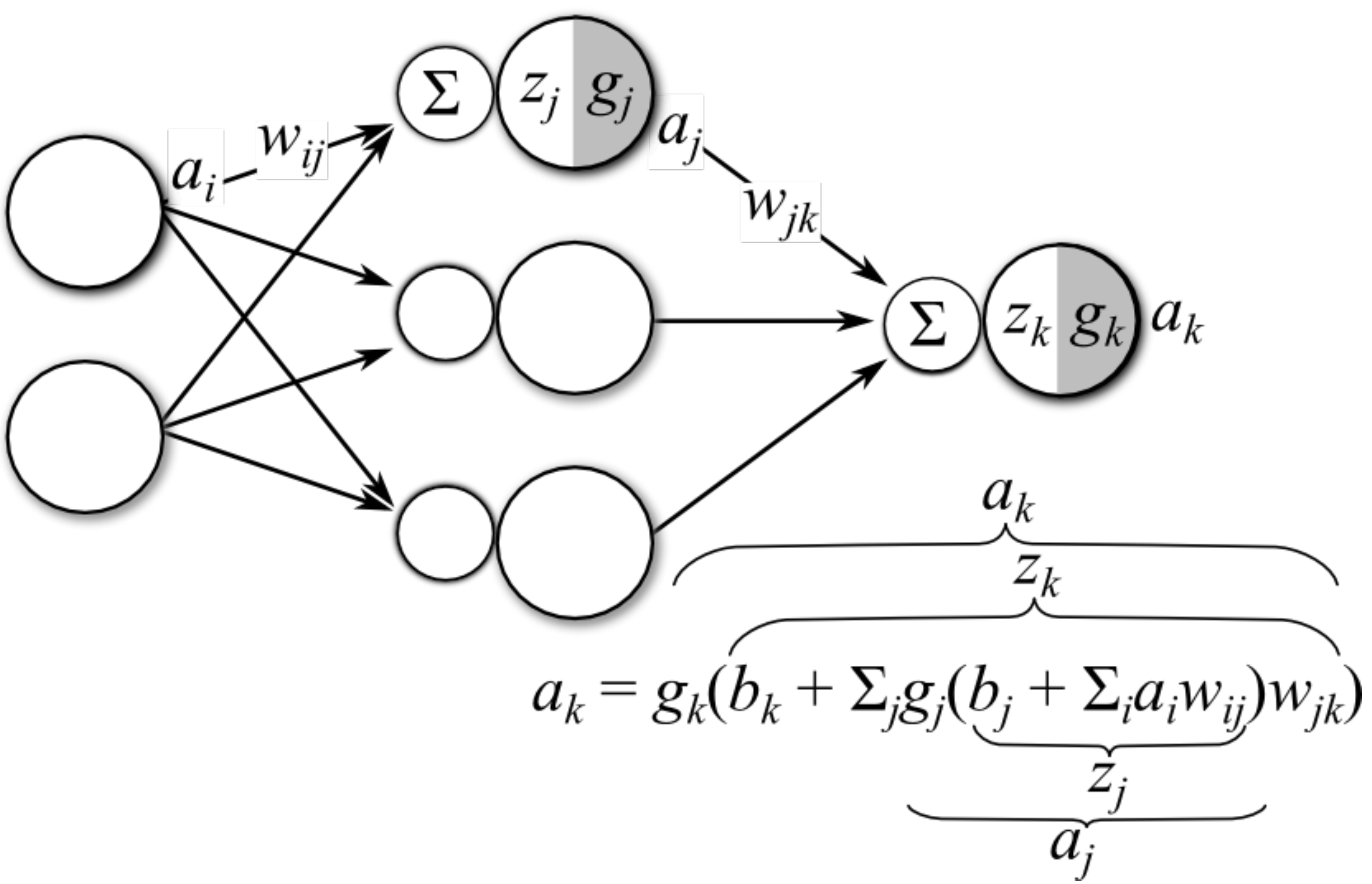


ARTIFICIAL NEURAL NETWORKS

* Problem: this network is not much more "expressive" than the Perceptron
  * It still can't XOR
* Solution: stack it! add a "hidden layer"



ARTIFICIAL NEURAL NETWORKS



w = weights

a = activations

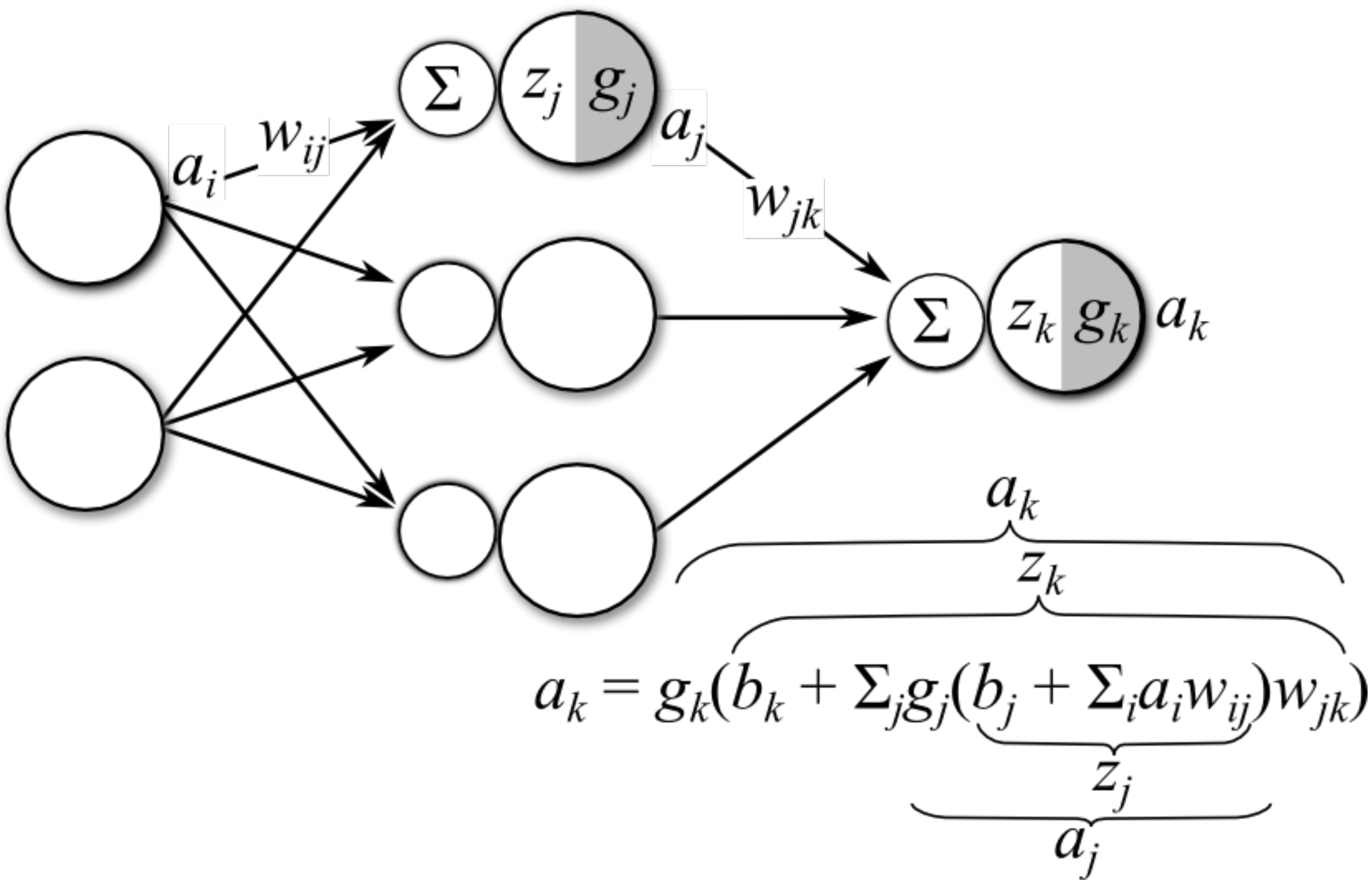
z = total input

g = activ. fxn

b = bias

h/t Dustin Stansbury

ARTIFICIAL NEURAL NETWORKS



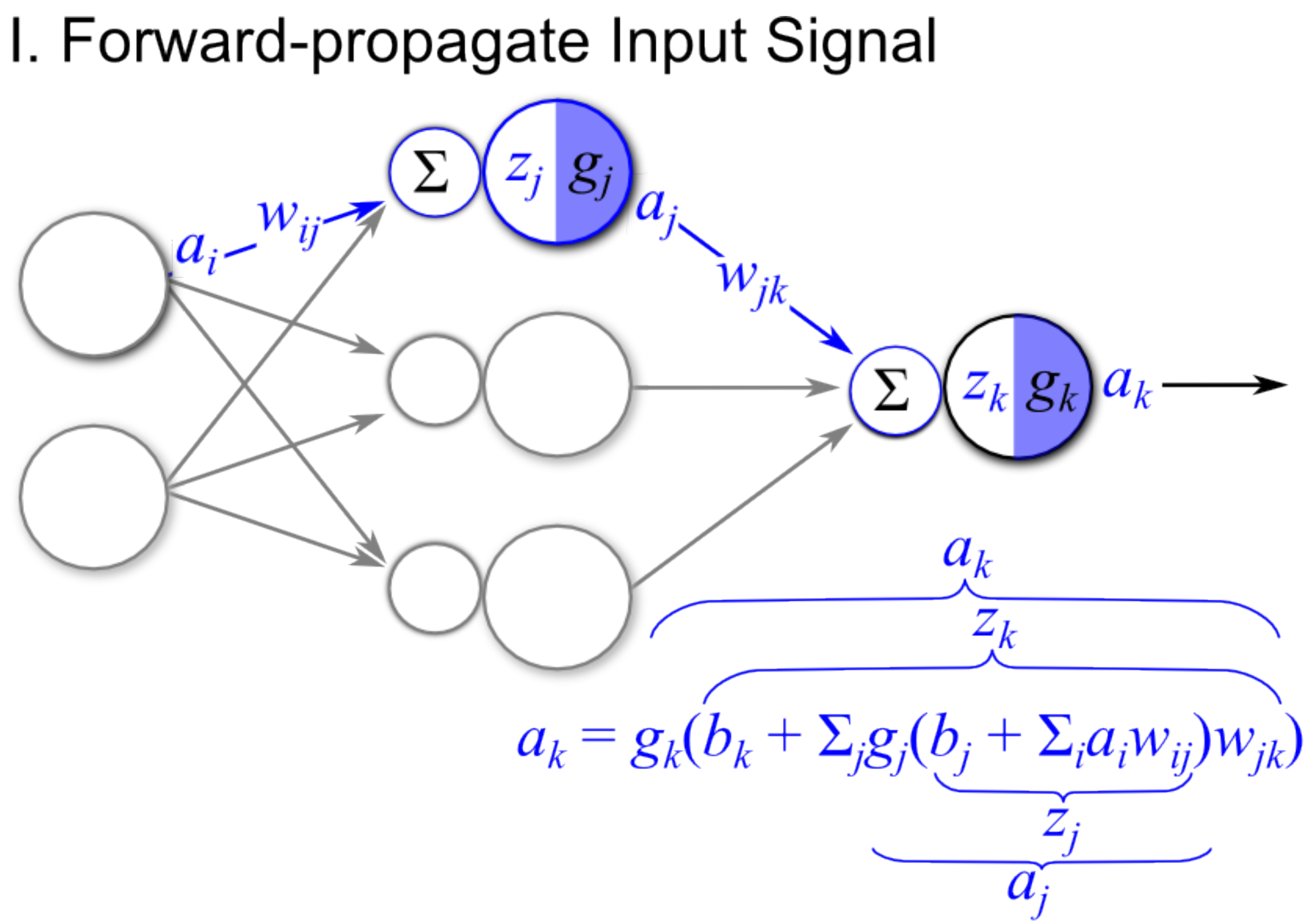
* How do we train this beast?

* Differentiate the loss function, same as ever!

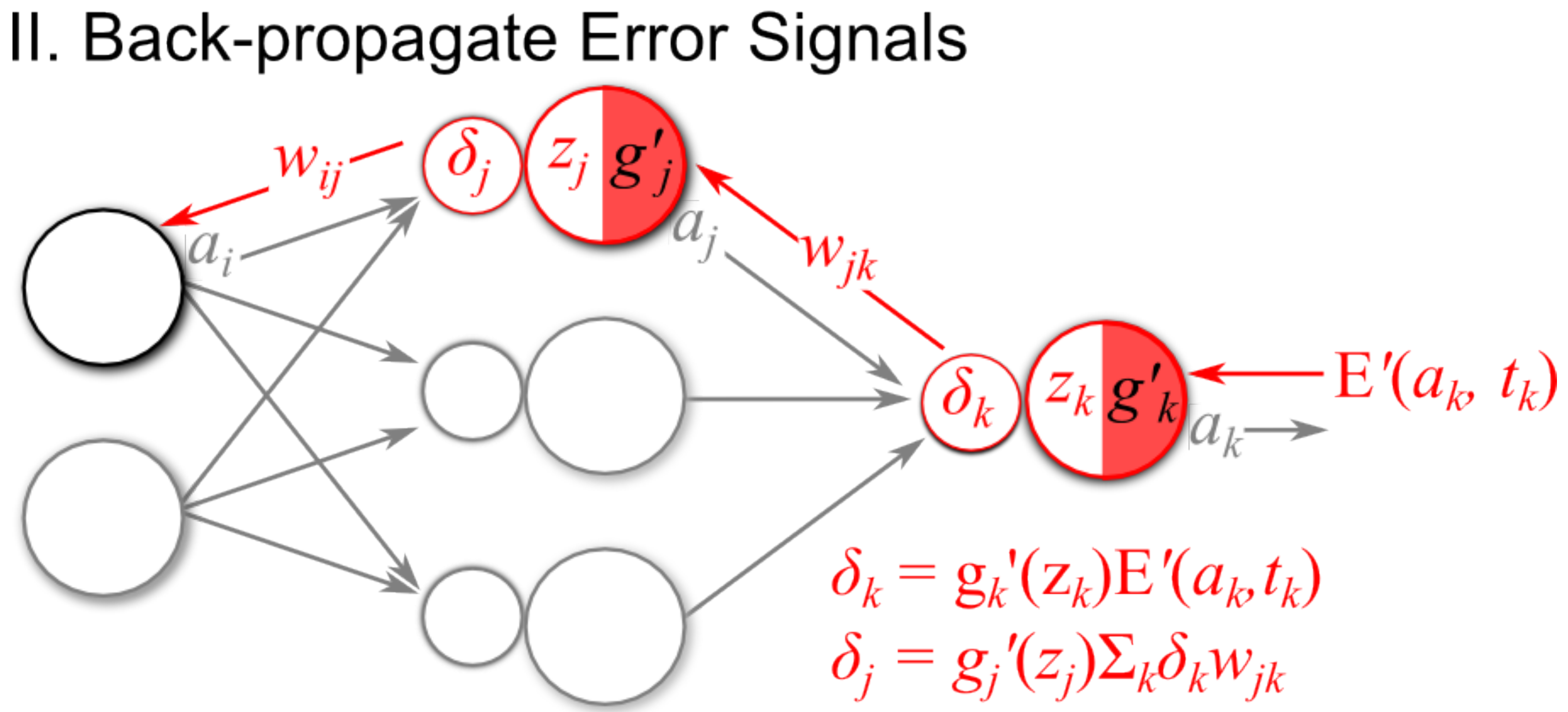
GRADIENT BACKPROPAGATION

* (Derivation of backpropagation)

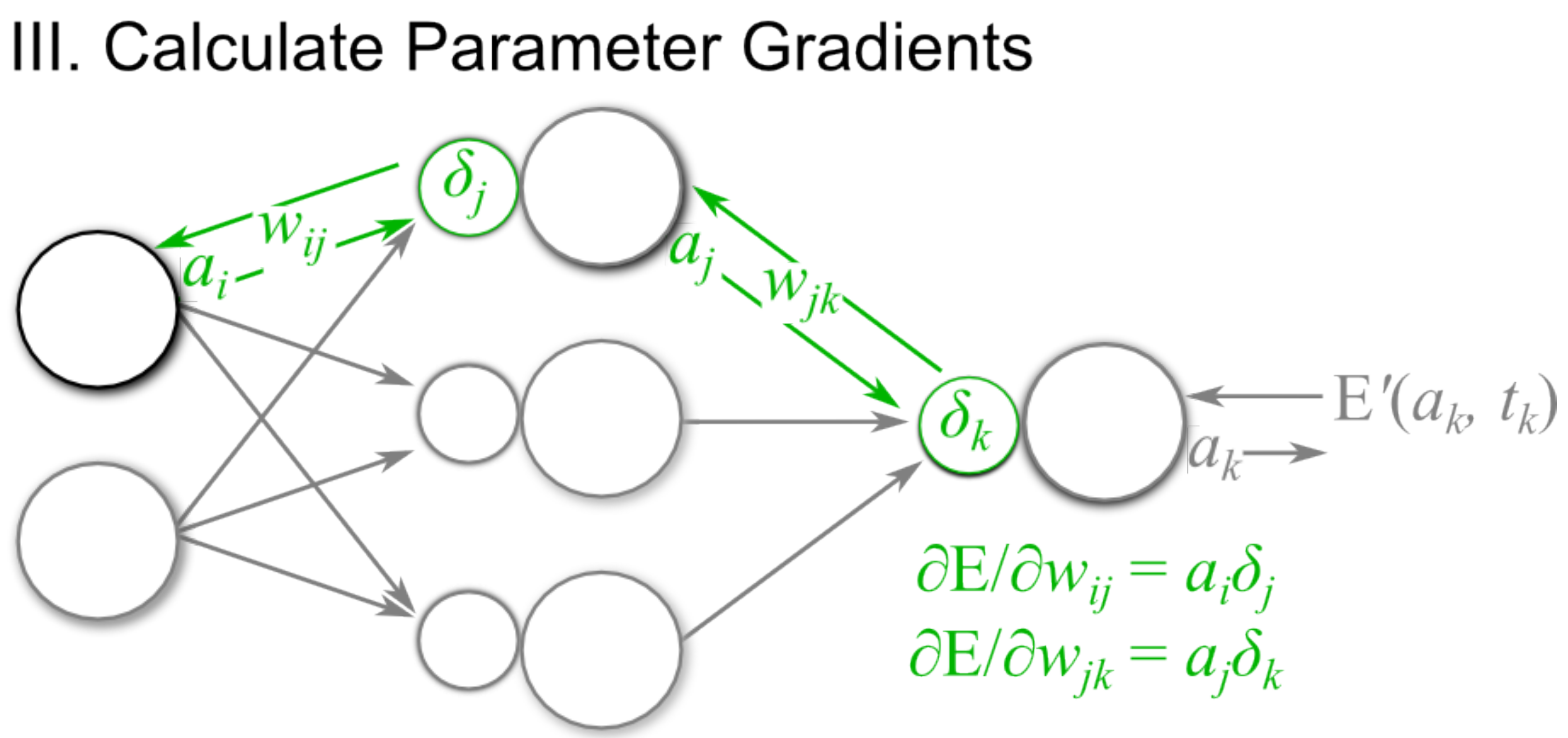
GRADIENT BACKPROPAGATION



GRADIENT BACKPROPAGATION



GRADIENT BACKPROPAGATION



GRADIENT BACKPROPAGATION

IV. Update Parameters

$$w_{ij} = w_{ij} - \eta(\partial E/\partial w_{ij})$$

$$w_{jk} = w_{jk} - \eta(\partial E/\partial w_{jk})$$

for learning rate η

GRADIENT BACKPROPAGATION

* Backprop enables **efficient** computation of gradients in networks of **arbitrary depth** by caching intermediate values δ

* It has probably been discovered in some form a few times in different fields

* In this field we often credit Rumelhart, Hinton, & Williams (1986)

ARTIFICIAL NEURAL NETWORKS

* What can networks with hidden layers do?

* **Universal approximation theorem:** These networks can do *literally anything* (as long as there is a non-polynomial output nonlinearity)

  * See Cybenko (1989) paper attached to this lecture for proof

ARTIFICIAL NEURAL NETWORKS

* But: it is not necessarily true that every function is ***learnable*** using backprop

* Different network architectures (e.g. different numbers of hidden layers & units per layer) are good at learning different functions

NEXT TIME

* Artificial neural networks for solving vision problems (e.g. categorization)