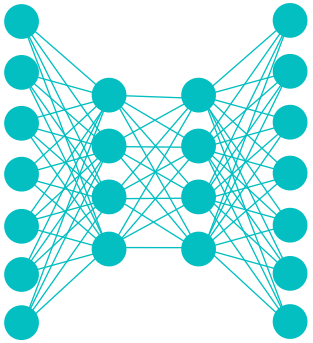
Lecture Notes for

Deep Learning II

Course Introduction
Lecture: Deep Learning Review



Logistics and Agenda

- Logistics
  - This class evolves across semesters (drastically!)
    - First offered in 2019
  - Using Canvas
  - Office Hours: after class, Mondays
  - GitHub: Basically one repository
  - Job Posting: Audio Annotation
- Agenda
  - Syllabus and Introductions
  - Presentation Selection
  - Topic Review

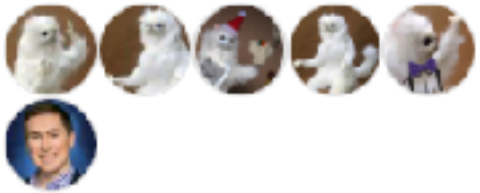
Syllabus

- Course Schedule 🤭
- Reading / Distance
- GitHub
- Grading
  - Labs x4 (60%)
  - Final Presentation and Project x1 (30%)
    - Could be replaced with comprehensive exam
  - Participation (Pass/Fail)
  - Class Presentation Summary (10%)
- LLM Usage

8000net

This organization houses a number of repositories for Dr. Larson's 8000 Level Neural Networks Course, Offered at SMU

People



Presenting

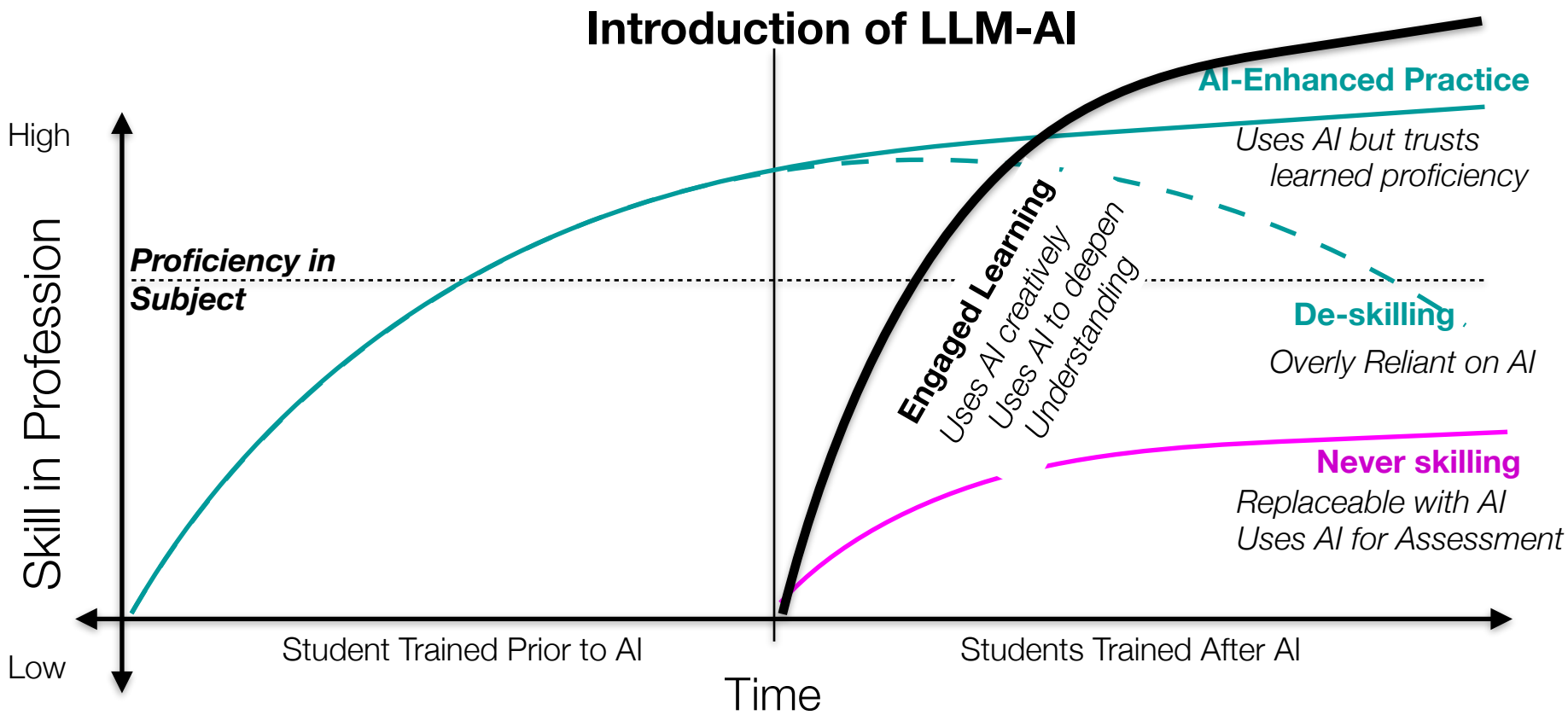
- First Presentation is coming up soon!
- During Semester: 12 Presentations Total (as a team)
- First Presentation → **Who wants to go first?**
  - ~10-15 Minutes (~5 mins per group member)
  - Present the topic of the paper, summarize important aspects
  - Make about 5 Visuals
    - typically: Slides
    - AND/OR Handouts
    - AND/OR Notebooks

Introductions

- Name, Majors, and Designation
- When you took CS5/7324
- The Topic in this course you are most excited about
- Something true or false about you

- Do NOT forget:
  - Pick out papers on Canvas

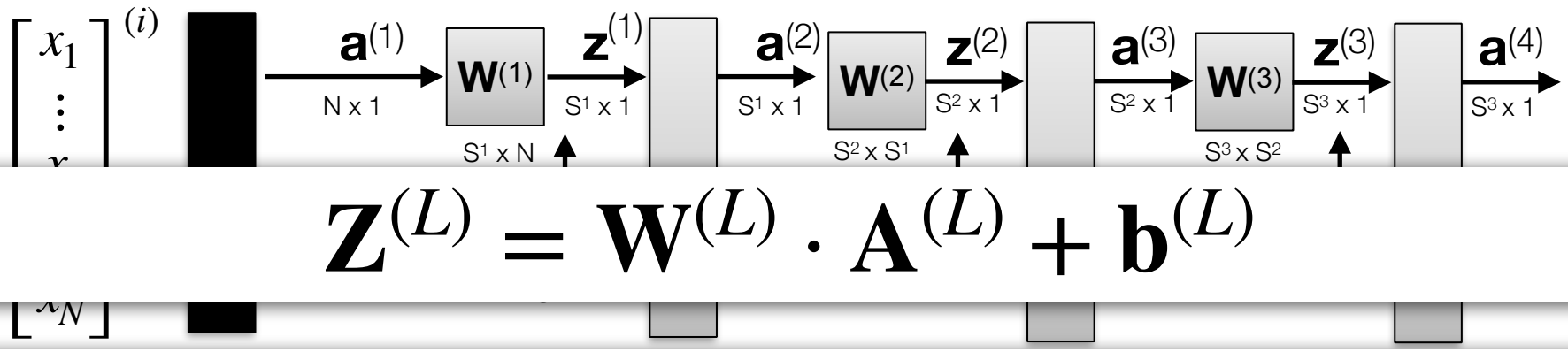
DL Review


Introduction of LLM-AI
AI-Enhanced Practice
Uses AI but trusts learned proficiency
High
Skill in Profession
Proficiency in Subject
Engaged Learning
Uses AI creatively
Uses AI to deepen Understanding
De-skilling
Overly Reliant on AI
Never skilling
Replaceable with AI
Uses AI for Assessment
Low
Student Trained Prior to AI
Students Trained After AI
Time

Prerequisite Topics, Review

- Feedforward networks and back propagation
- Optimization methods (SGD, AdaM)
- Computation graph and auto differentiation
- Embedding layers and multi-modal input
- Convolutional networks
- Sequential networks with emphasis on transformers
  - some transformer review will happen later in semester

Neural Network Structure



$$\mathbf{Z}^{(L)} = \mathbf{W}^{(L)} \cdot \mathbf{A}^{(L)} + \mathbf{b}^{(L)}$$

$$\mathbf{Z}^{(L)} = \mathbf{W}^{(L)} \cdot \phi(\mathbf{Z}^{(L-1)}) + \mathbf{b}^{(L)}$$

$$\left[\mathbf{z}^{(L)}\right]^{(i)} = \mathbf{W}^{(L)} \cdot \left[\mathbf{a}^{(L)}\right]^{(i)} + \mathbf{b}^{(L)}$$

$$\begin{bmatrix} z_1^{(L)} \\ z_2^{(L)} \\ \vdots \\ z_{S^L}^{(L)} \end{bmatrix} = \mathbf{W}^{(L)} \cdot \begin{bmatrix} a_1^{(L)} \\ \vdots \\ a_{S^{L-1}}^{(L)} \end{bmatrix} + \mathbf{b}^{(L)}$$

b is broadcast added

$$\begin{bmatrix} \begin{bmatrix} z_1^{(L)} \\ z_2^{(L)} \\ \vdots \\ z_{S^L}^{(L)} \end{bmatrix}^{(1)}, \begin{bmatrix} z_1^{(L)} \\ z_2^{(L)} \\ \vdots \\ z_{S^L}^{(L)} \end{bmatrix}^{(2)} \cdots \begin{bmatrix} z_1^{(L)} \\ z_2^{(L)} \\ \vdots \\ z_{S^L}^{(L)} \end{bmatrix}^{(M)} \end{bmatrix} = \mathbf{W}^{(L)} \cdot \begin{bmatrix} \begin{bmatrix} a_0^{(L)} \\ a_1^{(L)} \\ \vdots \\ a_{S^{L-1}}^{(L)} \end{bmatrix}^{(1)}, \begin{bmatrix} a_0^{(L)} \\ a_1^{(L)} \\ \vdots \\ a_{S^{L-1}}^{(L)} \end{bmatrix}^{(2)} \cdots \begin{bmatrix} a_0^{(L)} \\ a_1^{(L)} \\ \vdots \\ a_{S^{L-1}}^{(L)} \end{bmatrix}^{(M)} \end{bmatrix} + \mathbf{b}^{(L)}$$

Adaptive Optimization

Adjust each element of gradient by the steepness (for each layer):

- AdaGrad

$$\rho_k = \frac{1}{\sqrt{\mathbf{G}_k + \epsilon}} \odot \nabla J(\mathbf{W}_k)$$

all operations are per element

where

$$\mathbf{G}_k = \gamma \cdot \mathbf{G}_{k-1} + \nabla J(\mathbf{W}_k) \odot \nabla J(\mathbf{W}_k)$$

- RMSProp

$$\rho_k = \frac{1}{\sqrt{\mathbf{V}_k + \epsilon}} \odot \nabla J(\mathbf{W}_k)$$

all operations are per element

$$\mathbf{G}_k = \nabla J(\mathbf{W}_k) \odot \nabla J(\mathbf{W}_k)$$

$$\mathbf{V}_k = \gamma \cdot \mathbf{V}_{k-1} + (1-\gamma) \cdot \mathbf{G}_k$$

- AdaDelta

$$\rho_k = \frac{\mathbf{M}_k}{\sqrt{\mathbf{V}_k + \epsilon}}$$

all operations are per element

$$\mathbf{M}_{k+1} = \gamma \cdot \mathbf{M}_k + (1-\gamma) \cdot \nabla J(\mathbf{W}_k)$$

- AdaM

update momentum

$$\mathbf{M}_{k+1} \leftarrow \beta_1 \cdot \mathbf{M}_k + (1-\beta_1) \cdot \nabla J(\mathbf{W}_k)$$

$$\hat{\mathbf{M}}_k \leftarrow \frac{\mathbf{M}_k}{(1-[\beta_1]^k)}$$

normalizer momentum

$$\mathbf{V}_{k+1} \leftarrow \beta_2 \cdot \mathbf{V}_k + (1-\beta_2) \cdot \nabla J(\mathbf{W}_k) \odot \nabla J(\mathbf{W}_k)$$

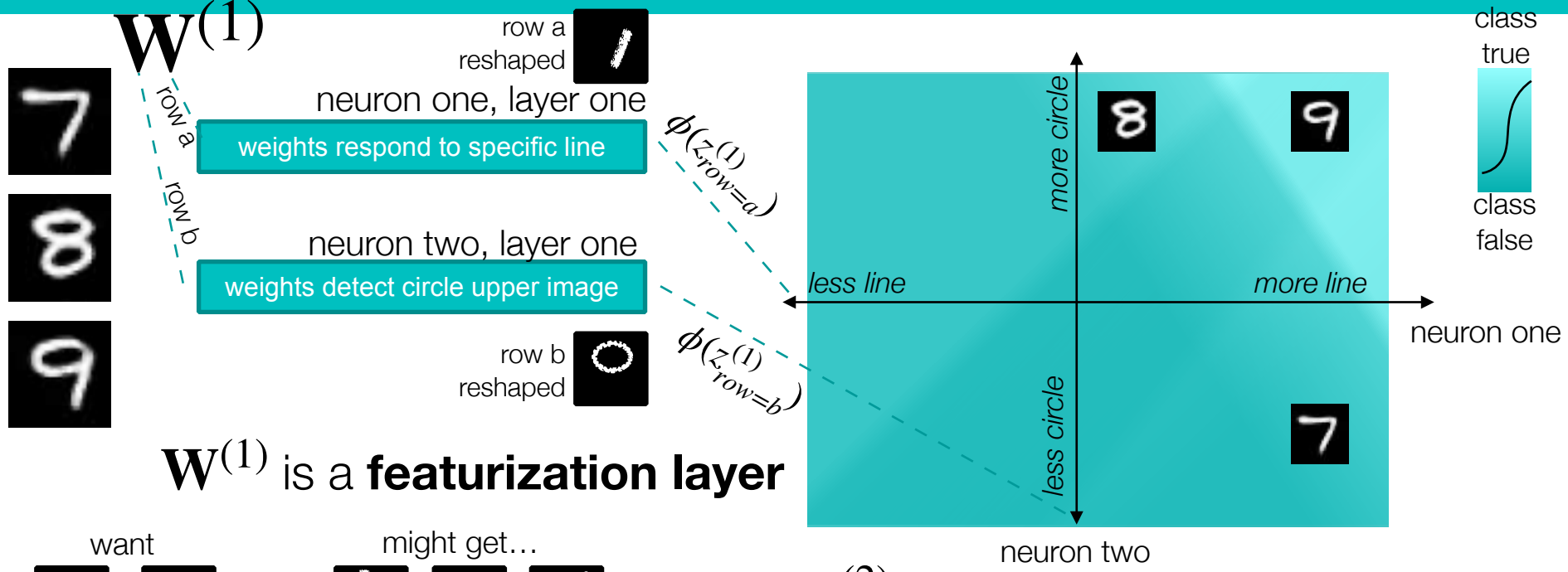
$$\hat{\mathbf{V}}_k \leftarrow \frac{\mathbf{V}_k}{(1-[\beta_2]^k)}$$

exploitation, boosting

full update

$$\mathbf{W}_{k+1} \leftarrow \mathbf{W}_k - \eta \cdot \frac{\hat{\mathbf{M}}_k}{\sqrt{\hat{\mathbf{V}}_k + \epsilon}}$$

Universality

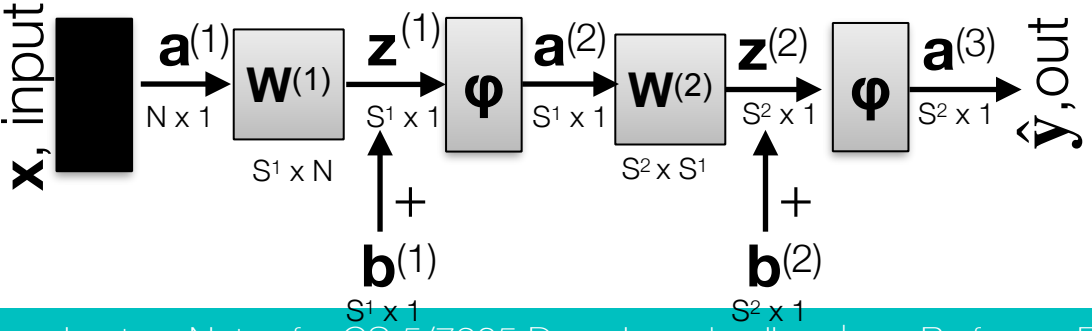


$\mathbf{W}^{(1)}$ is a **featurization layer**

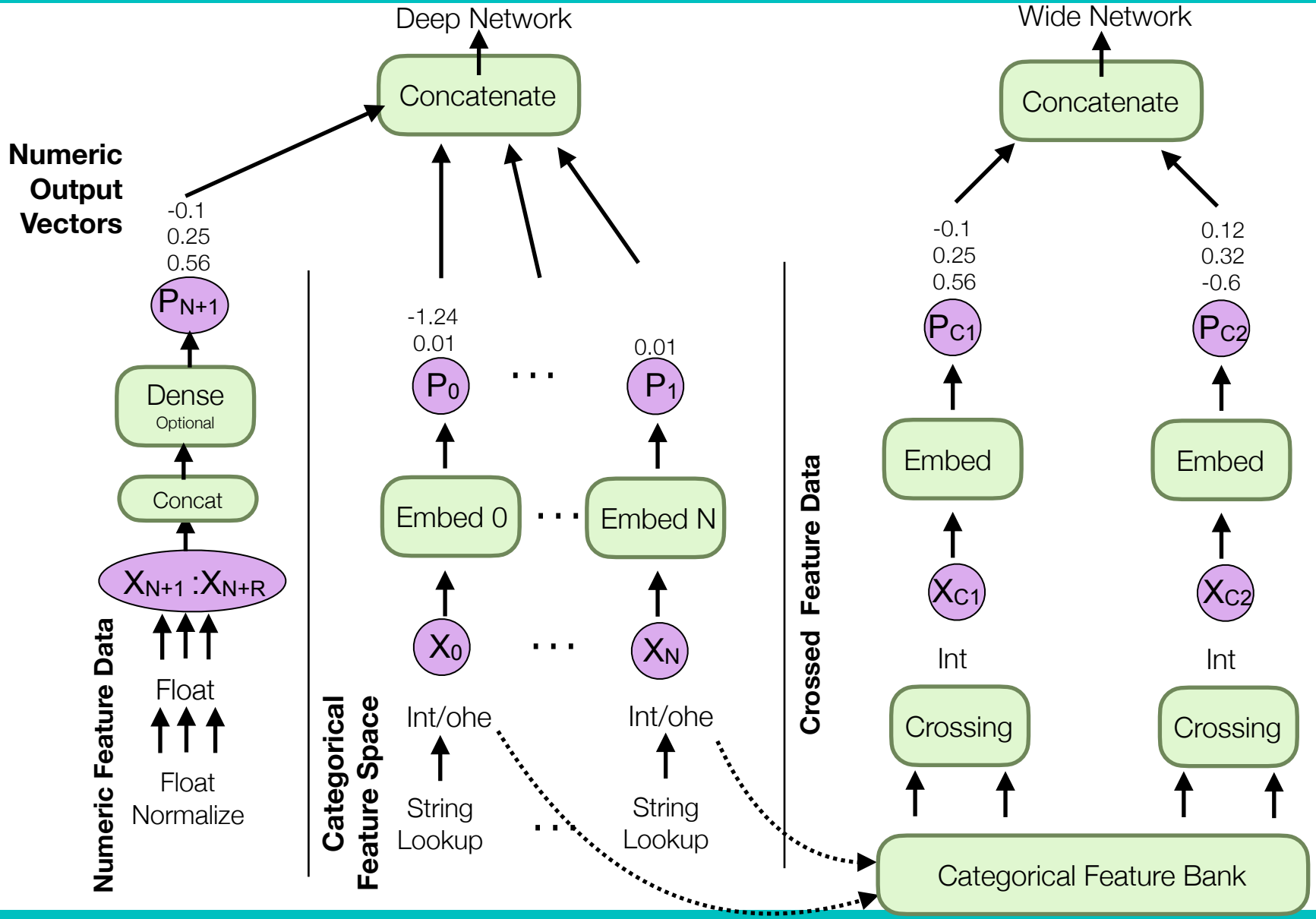
adding layers and convolution help obtain desired behavior…

$\mathbf{W}^{(2)}$ is **classification layer,** like one versus all logistic regression

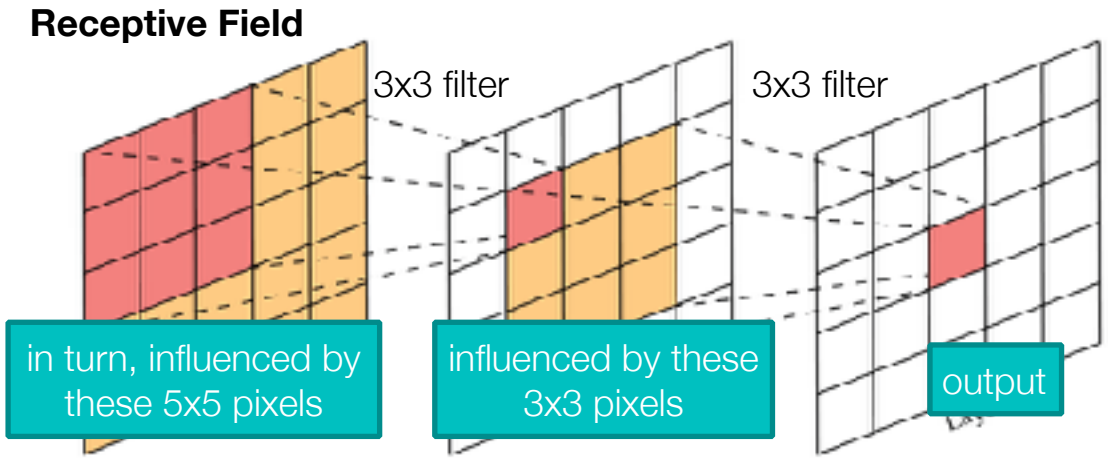
- One nonlinear hidden layer with an output layer can ***perfectly train any problem with enough data***, but might be memorizing…
- … could be better to have ***even more layers*** for more generalizing features

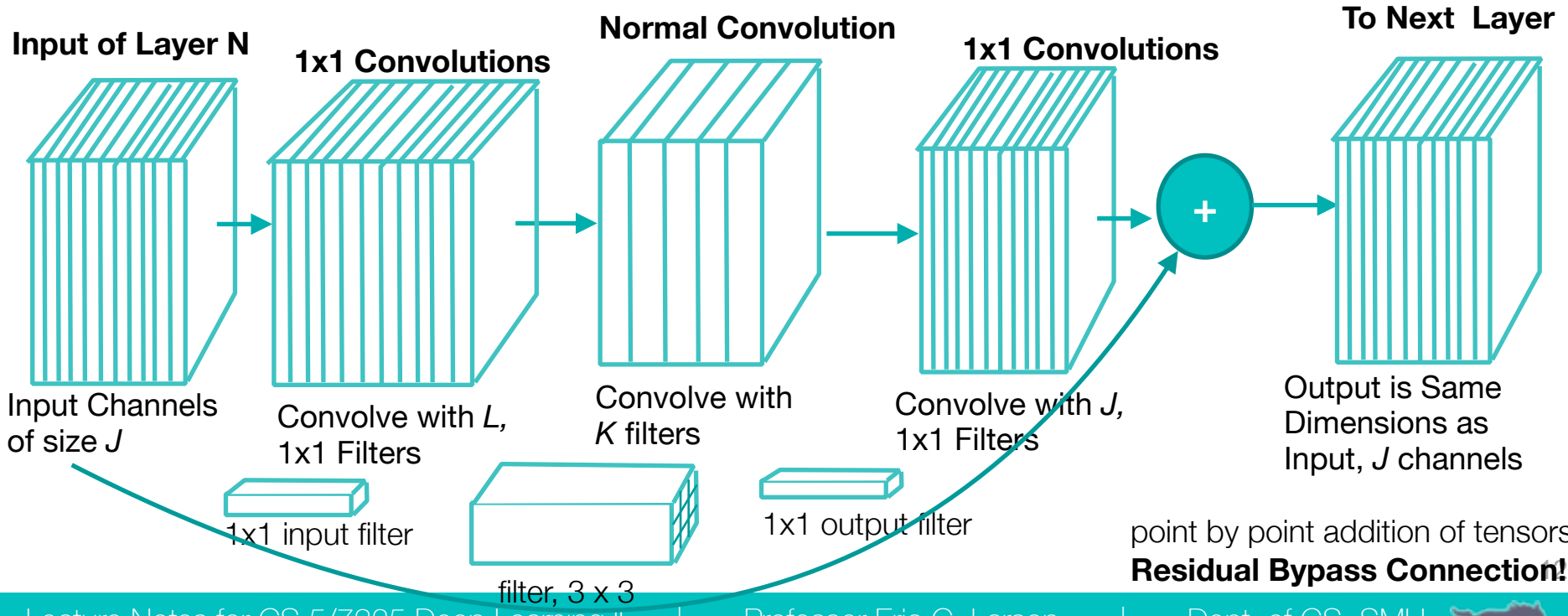


Computation Graph, Feature Spaces



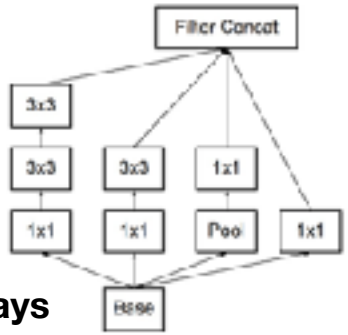
Convolutional Networks



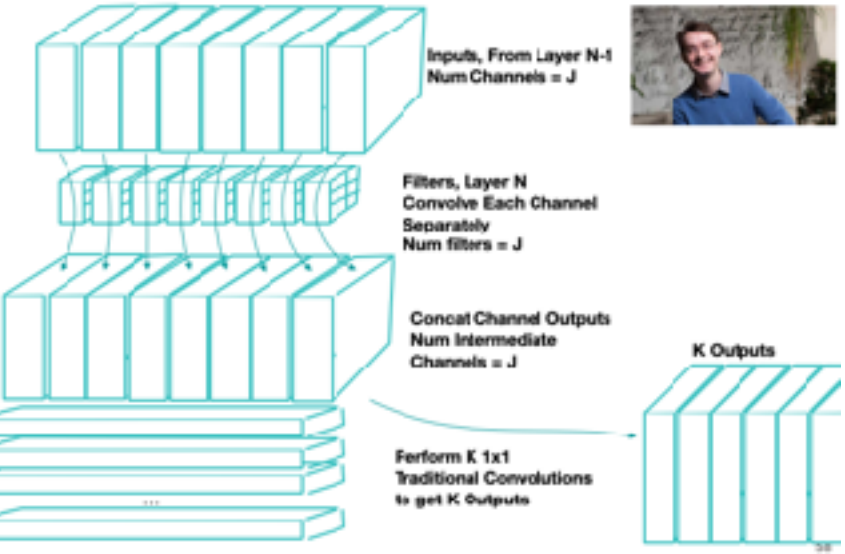


CNN Techniques

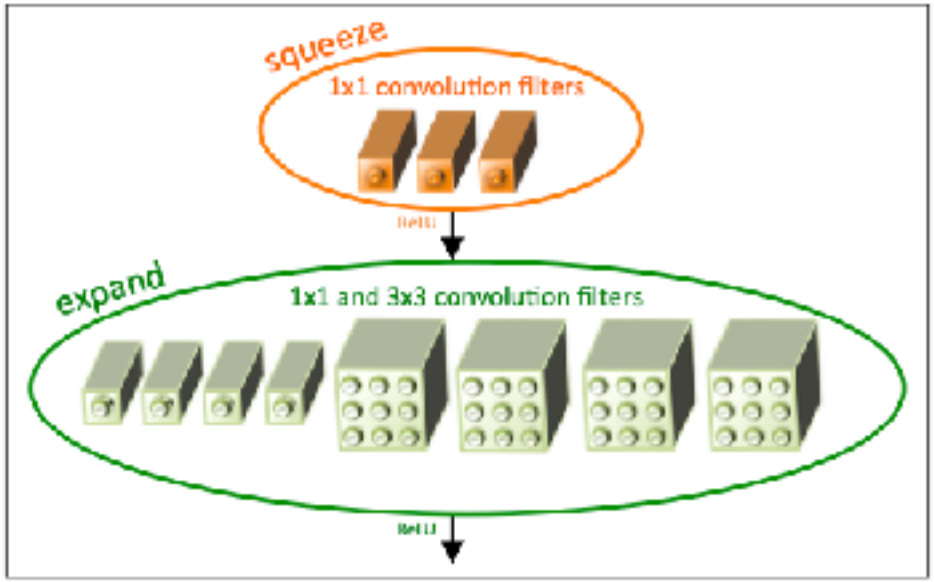
- Bottlenecks (1x1 filters)
- Parallel Paths, Concatenation
- Residual Paths
- Separable CNN



Parallel Pathways



Separable Convolution



Squeeze

$$SR = \frac{|F_{s1x1}|}{|F_{e1x1}| + |F_{e3x3}|}$$

Controls how much to bottleneck

$$PCT_{3x3} = \frac{|F_{e3x3}|}{|F_{e1x1}| + |F_{e3x3}|}$$

Controls num filter params
(how many 1x1 versus 3x3)

Lecture Notes for

Deep Learning II

Course Introduction
Lecture: Deep Learning Review