Lecture 17: Exam 1 Review

Mark Hasegawa-Johnson, 2/2024

These slides are public domain: Copy at will

Outline

- How to take the exam
- What’s on the exam
- Sample problems

How to take the exam

- Go to https://cbtf.Illinois.edu, choose CS440/ECE448 exam 1, and register the time and location where you will take the exam
- If you can’t register at cbtf.Illinois.edu, contact us on campuswire BEFORE 1pm Monday

How to take the exam

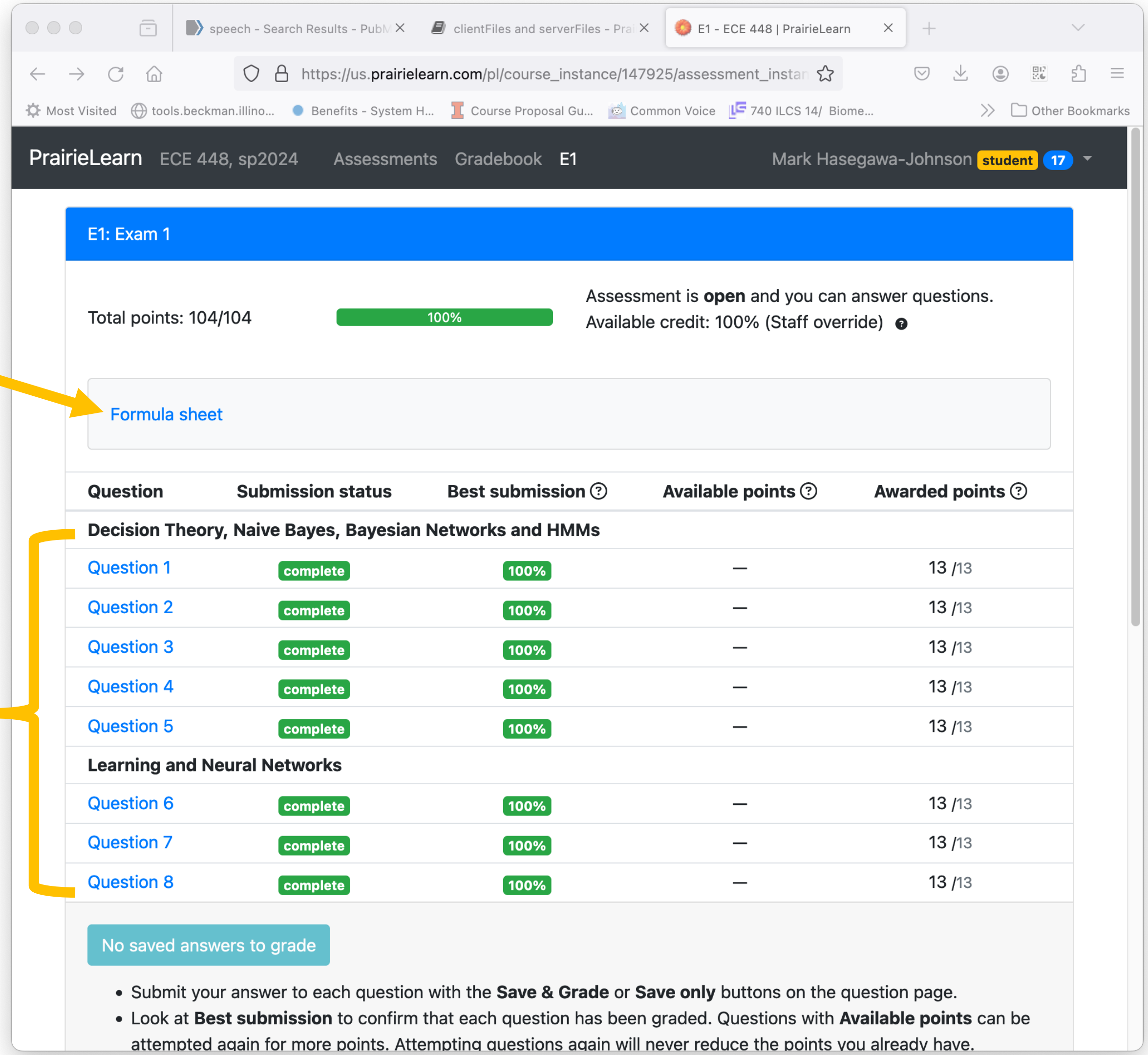
- Show up on time at the appointed location! It's possible to reschedule if you miss an exam, but it will require you to go to a CBTF location to apply in person for permission to do so.
- Bring: Pencils and erasers
- CBTF will provide: Scratch paper
- The exam will have attached: a PDF formula sheet, which is also available now on the course web page, so you can see what will be on it

Exam format

- The exam will have 8 questions, each worth 13 points --- 104 points total. If you get 104 out of 104, that is worth 100%.
  - 100/104 = 96%.
- Each question will be multiple choice, very similar to the daily quiz questions in class
- You will get full credit (13 points) if you get it right on the first try
- If not, try again (and again and again and again, if necessary), for reduced credit
  - E.g., if a question has 6 options, then you get 13,9,7,5,3,1 points for getting the correct answer on the 1st/2nd/3rd/4th/5th/6th try.

Exam format

- Formula sheet
- Questions to answer



Outline

- How to take the exam
- What's on the exam
- Sample problems

Topics

- Bayesian networks – 5 questions
  - probability, decision theory, naïve Bayes, Bayes nets, HMMs, fairness
- Neural networks – 3 questions
  - learning, regression, perceptron, softmax, multilayer, vision, CNN

Probability

- Probability Mass and Probability Density

$$P(X = x) = \Pr(X = x) \quad \dots \quad \text{or} \quad \dots \quad P(X = x) = \frac{d}{dx}\Pr(X \le x)$$

- Jointly random variables

$$P(X = \boldsymbol{x}) = P(X_1 = x_1, \cdots, X_n = x_n)$$

- Conditional Probability and Independence

$$P(X, Y) = P(X|Y)P(Y)$$
$$P(X|Y) = P(X) \Leftrightarrow P(X, Y) = P(X)P(Y)$$

- Expectation

$$E[f(X, Y)] = \sum_{x,y} f(x, y)P(X = x, Y = y) \quad \dots \quad \text{or} \quad \dots$$
$$E[f(X, Y)] = \iint_{-\infty}^{\infty} f(x, y)P(X = x, Y = y)dxdy$$

- Mean, Variance and Covariance

$$\Sigma = E[(X - E[X])(X - E[X])^T]$$

Decision theory

- Minimum Probability of Error = Maximum *A Posteriori:*
$$f(x) = \underset{y}{\operatorname{argmax}}\, P(Y = y|X = x)$$
- Bayes Error Rate:
$$\text{Bayes Error Rate} = \sum_x P(X = x) \min_y P(Y \neq y|X = x)$$
- Confusion Matrix, Precision & Recall, Sensitivity & Selectivity
$$\text{Precision} = P(Y = 1|f(X) = 1) = \frac{TP}{TP + FP}$$
$$\text{Sensitivity} = \text{Recall} = P(f(X) = 1|Y = 1) = \frac{TP}{TP + FN}$$
$$\text{Selectivity} = P(f(X) = 0|Y = 0) = \frac{TN}{TN + FP}$$
- Train, Dev, and Test Corpora

Naïve Bayes

- MPE = MAP with Bayes' rule:

$$f(x) = \underset{y}{\operatorname{argmax}}(\log P(Y=y) + \log P(X=x|Y=y))$$

- naïve Bayes:

$$\log P(X=x|Y=y) \approx \sum_{i=1}^{n} \log P(W=w_i|Y=y)$$

- maximum likelihood parameter estimation:

$$P(W=w_i|Y=y) = \frac{\text{Count}(w_i, y)}{\sum_{v\in V} \text{Count}(v, y)}$$

- Laplace Smoothing:

$$P(W=w_i|Y=y) = \begin{cases} \dfrac{k + \text{Count}(w_i, y)}{k + \sum_{v\in V}(k + \text{Count}(v, y))} & W = OOV \text{ is possible} \\ \dfrac{k + \text{Count}(w_i, y)}{\sum_{v\in V}(k + \text{Count}(v, y))} & \text{otherwise} \end{cases}$$

Bayesian Networks

- Bayesian classifier: $f(x) = \underset{y}{\operatorname{argmax}} P(Y = y|X = x)$
- Bayesian network: A better way to represent knowledge
  - Each variable is a node.
  - An arrow between two nodes means that the child depends on the parent.
- Inference using a Bayesian network

$$P(B = \top, J = \top) = \sum_{e=\top}^{\bot} \sum_{a=\top}^{\bot} P(B = \top)P(E = e)P(A = a|B = \top, E = e)P(J = \top|A = a)$$

- Key ideas: Independence and Conditional independence
  - Independent = no common ancestors
  - Conditionally independent = (1) no common descendants, and (2) none of the descendants of one are ancestors of the other

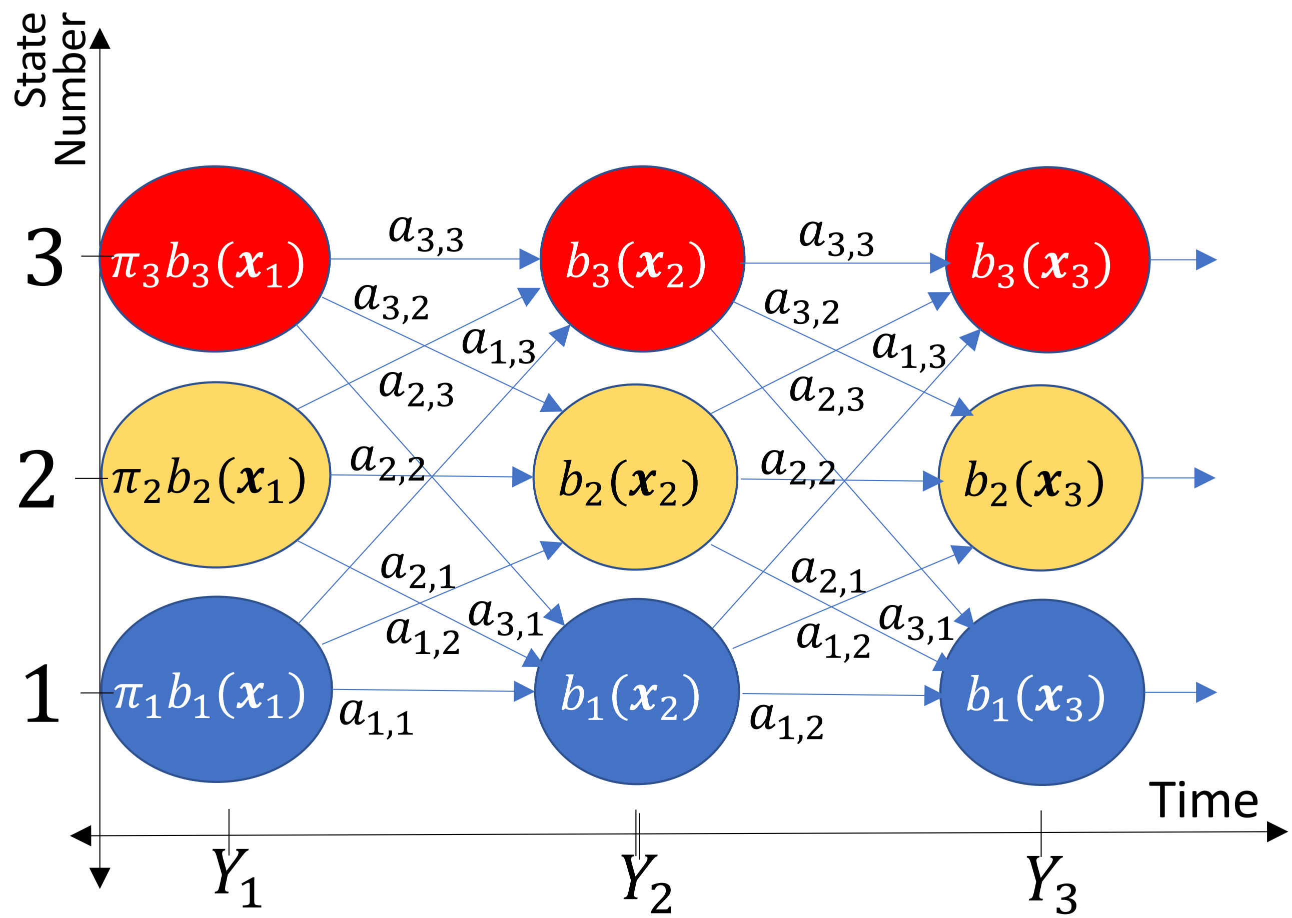
HMM: Viterbi Algorithm

Trellis is used to find the most likely path.

For example:

$$v_t(j) = \max_i v_{t-1}(i)\, a_{i,j} b_j(\boldsymbol{x}_t)$$

$$\psi_t(j) = \operatorname*{argmax}_i v_{t-1}(i)\, a_{i,j} b_j(\boldsymbol{x}_t)$$



Three Definitions of Fairness

- Demographic Parity: $P(f(X)|A = 1) = P(f(X)|A = 0)$
- Equal Odds: $P(f(X)|Y, A = 1) = P(f(X)|Y, A = 0)$
- Predictive Parity: $P(Y|f(X), A = 1) = P(Y|f(X), A = 0)$

Those three things can only all be true, all at the same time, if:

- $P(Y|A = 1) = P(Y|A = 0)$

Learning

- **Biological inspiration:** Neurons that fire together wire together. Given enough training examples (x_i, y_i), can we learn a desired function so that $f(x) \approx y$?
- **Classification tree:** Learn a sequence of if-then statements that computes $f(x) \approx y$
- **Mathematical definition of supervised learning:** Given a training dataset, $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, find a function f that minimizes the risk, $\mathcal{R} = \mathrm{E}[\ell(Y, f(X))]$.
- **Overtraining:** $\mathcal{R}_{\mathrm{emp}} = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$ reaches zero if you train long enough.
- **Early Stopping**: Stop when error rate on the dev set reaches a minimum

Linear Regression

- Definition of linear regression

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

- Mean-squared error

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i \,, \qquad \mathcal{L}_i = \frac{1}{2} \epsilon_i^2, \qquad \epsilon_i = f(\boldsymbol{x}_i) - y_i$$

- Gradient descent

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \boldsymbol{x}_i$$

- Stochastic gradient descent

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}}, \qquad \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}} = \epsilon_i \boldsymbol{x}_i$$

Perceptron

- Linear Classifiers: $f(\boldsymbol{x}) = \operatorname{argmax} \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$

- One-hot vectors: $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_v(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \mathbb{1}_{\operatorname{argmax} \boldsymbol{W}\boldsymbol{x}=1} \\ \vdots \\ \mathbb{1}_{\operatorname{argmax} \boldsymbol{W}\boldsymbol{x}=v} \end{bmatrix}, \ \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbb{1}_{y=1} \\ \mathbb{1}_{y=2} \\ \vdots \end{bmatrix}$

- Perceptron learning algorithm:

$$\boldsymbol{w}_c \leftarrow \begin{cases} \boldsymbol{w}_c - \eta\boldsymbol{x} & c = \operatorname{argmax} \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \\ \boldsymbol{w}_c + \eta\boldsymbol{x} & c = y \\ \boldsymbol{w}_c & \text{otherwise} \end{cases}$$

- Softmax: $f_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^T \boldsymbol{x} + b_c)}{\sum_{k=1}^{v} \exp(\boldsymbol{w}_k^T \boldsymbol{x} + b_k)} \approx \Pr(Y = c|\boldsymbol{x})$

- Sigmoid: $\sigma(\boldsymbol{w}^T \boldsymbol{x} + b) = \frac{1}{1+e^{-(\boldsymbol{w}^T \boldsymbol{x} + b)}} \approx \Pr(Y = 1|\boldsymbol{x})$

- Cross-entropy: $\mathcal{L} = -\ln f_y(\boldsymbol{x})$, $\frac{\partial \mathcal{L}}{\partial f_c(\boldsymbol{x})} = \begin{cases} -\frac{1}{f_c(\boldsymbol{x})} & c = y \\ 0 & \text{otherwise} \end{cases}$

- Gradient descent: $\boldsymbol{w}_c \leftarrow \boldsymbol{w}_c - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_c}$

- Derivative of the cross-entropy of a softmax:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_c} = \epsilon_c \boldsymbol{x}, \qquad \epsilon_c = \begin{cases} f_c(\boldsymbol{x}_i) - 1 & c = y \text{ (output should be 1)} \\ f_c(\boldsymbol{x}_i) - 0 & \text{otherwise(output should be 0)} \end{cases}$$

Multi-Layer

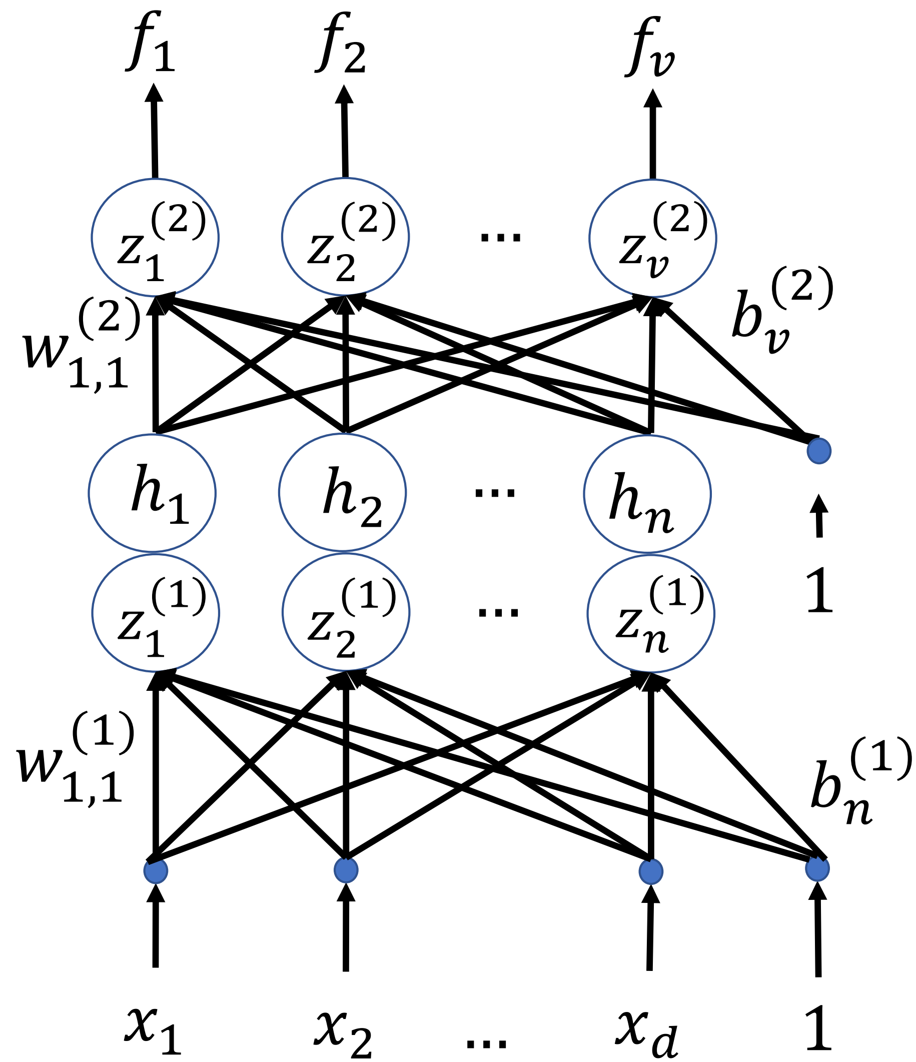
For example:

$$f_k = \underset{k}{\text{softmax}}\, \mathbf{z}^{(2)}$$

$$z_k^{(2)} = b_k^{(2)} + \sum_{j=1}^{n} w_{k,j}^{(2)} h_j$$

$$h_j = \text{ReLU}\left(z_j^{(1)}\right)$$

$$z_j^{(1)} = b_j^{(1)} + \sum_{i=1}^{d} w_{j,i}^{(1)} x_i$$



Back-propagation

For example, if the loss is cross-entropy, then

$$\frac{\partial \mathcal{L}}{\partial z_k^{(2)}} = f_k - \mathbb{1}_{y=k}$$

So the weight gradient is:

$$\frac{\partial \mathcal{L}}{\partial w_{j,i}^{(1)}} = \sum_{k=1}^{v} \left(\frac{\partial \mathcal{L}}{\partial z_k^{(2)}}\right)\left(\frac{\partial z_k^{(2)}}{\partial h_j}\right)\left(\frac{\partial h_j}{\partial w_{j,i}^{(1)}}\right)$$

$$= \sum_{k=1}^{v} (f_k - \mathbb{1}_{y=k}) w_{k,j}^{(2)} \mathbb{1}_{h_j>0} x_i$$

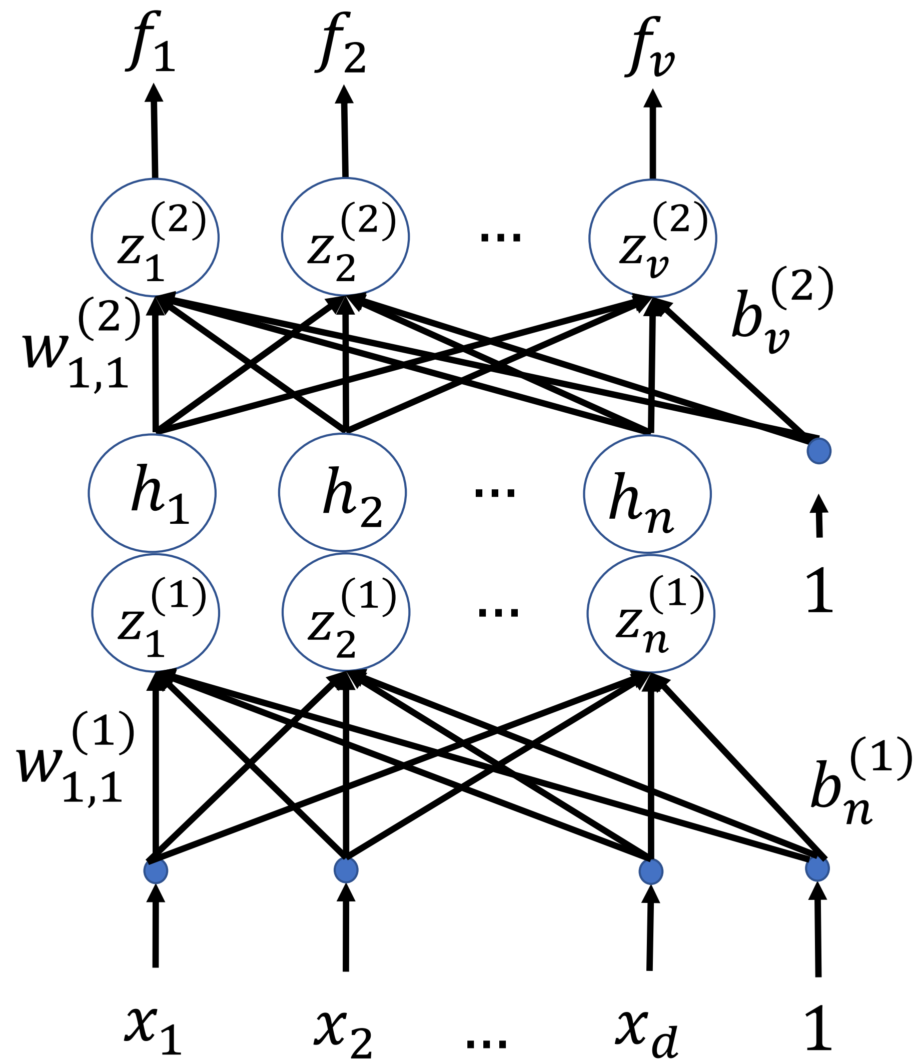


Image formation & processing

- Pinhole camera equations:

$$\frac{x'}{f} = -\frac{x}{z}, \qquad \frac{y'}{f} = -\frac{y}{z}$$

- Vanishing point = parallel lines:

$$x_1 = az + c_1, \qquad y_1 = bz + d_1$$
$$x_2 = az + c_2, \qquad y_2 = bz + d_2$$

- Edge detection using difference-of-Gaussians:

$$h(m,n) = \frac{1}{2\pi\sigma^2} e^{-\left(\left(\frac{m}{\sigma}\right)^2 + \left(\frac{n}{\sigma}\right)^2\right)}$$

$$h_x'(x', y') = \frac{(h(x'+1, y') - h(x'-1, y'))}{2}$$

$$h_y'(x', y') = \frac{(h(x', y'+1) - h(x', y'-1))}{2}$$

Convolution and Max Pooling

$$y[k,l] = h[k,l] * x[k,l] = \sum_i \sum_j x[k-i, l-j] h[i,j]$$

$$\frac{d\mathcal{L}}{dh[i,j]} = \sum_k \sum_l \frac{d\mathcal{L}}{dy[k,l]} \frac{dy[k,l]}{dh[i,j]}$$

$$z[m,n] = \max_{\substack{(m-1)p+1 \le k \le mp, \\ (n-1)p+1 \le l \le np}} y[k,l]$$

$$\frac{d\mathcal{L}}{dy[k,l]} = \begin{cases} \dfrac{d\mathcal{L}}{dz[m,n]} & \text{if } y[k,l] = \max\limits_{\substack{(m-1)p+1 \le i \le mp, \\ (n-1)p+1 \le j \le np}} y[i,j] \\ 0 & \text{otherwise} \end{cases}$$

Outline

- How to take the exam
- What's on the exam
- Sample problems

Sample problems

- You can do the daily quizzes again, as often as you like
  - Only your highest score counts toward your grade (presumably this is the one you received the first time you did the quiz, when 100% was possible)
  - After you get a problem correct, you can click “Try a new variant” to try a new variant
- Sample exam is available
  - Coverage is similar to the exam next week
  - Style of questions is quite different

Sample problems

- Sample problems
  - NOT the same format as the real exam
- Formula sheet
  - SAME as the real exam