Final Exam Review

Exam:

Cumulative

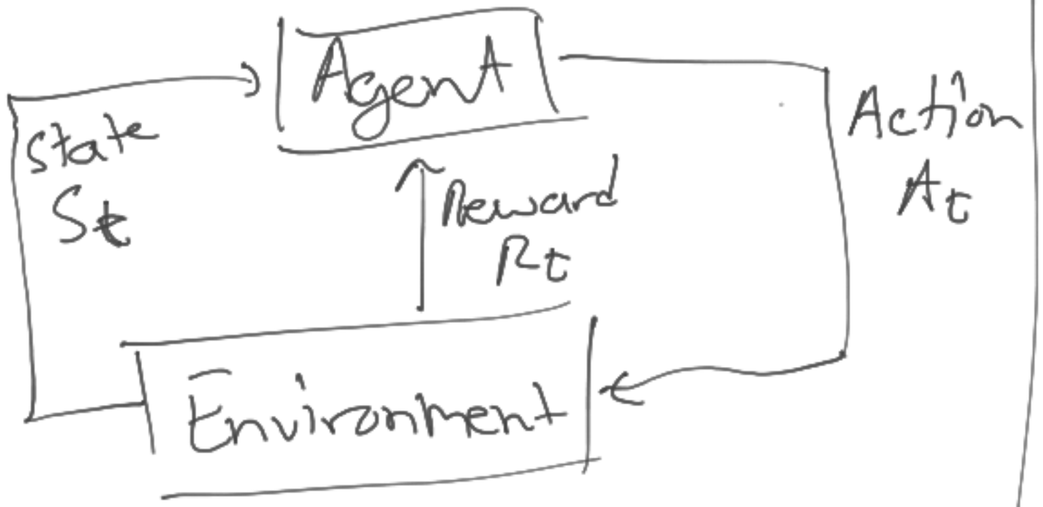
Completable in 2 hour exam slot. (May 11, 2pm-4pm)

I will be available during 2 hour exam slot

Posted 12:01 am May 11, due 11:59 pm May 11.

Approximately same format as midterm.

More focus on R2 than other topics.



$t \in \{0, 1, 2, \ldots\}$ (time step)

$\mathcal{S}$ = set of possible states

$\mathcal{A}$ = set of possible actions

S_t = state at time t

A_t = action at time t

R_t = reward at time t

s = some state

a = some action.

$\pi(s,a) = \Pr(A_t = a \mid S_t = s), \ \forall t$ ← policy

$J(\pi)$ = how good is π? $= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,;\, \pi\right]$ ← remember, generated using π. ; = semi-colon. $\mathbb{E}$ = expected value. → bold E.

$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid \pi\right]$

γ = reward discount parameter. $\gamma \in [0,1]$.

$\pi^* \in \arg\max_{\pi \in \Pi} J(\pi)$ ← in, not = ; optimal policy ; set of all possible policies

episode = one run, starting from $t=0$

θ = policy parameters (vector).

$\pi_\theta(s,a) = \Pr(A_t = a \mid S_t = s \,;\, \theta)$ ← semi-colon.

→ softmax action selection: $\pi_\theta(s,a) = \dfrac{e^{\sum_j \theta_j^a \phi_j(s)}}{\sum_{a'} e^{\sum_j \theta_j^{a'} \phi_j(s)}}$

$\theta = [\theta^1, \theta^2, \ldots, \theta^{|\mathcal{A}|}]$

$\theta^a = [\theta_1^a, \theta_2^a, \ldots, \theta_{\text{numfeatures}}^a]$

(state) value function.

$v^\pi(s)$ = value of state s under policy π

$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \,;\, \pi\right]$

w = weights/parameters for value function estimate

v_w = estimate of v^π using weights w.

$v^\pi(s) \approx v_w(s) \quad \forall s \in \mathcal{S}$

$\delta_t = R_t + \gamma v_w^{\pi}(S_{t+1}) - v_w^{\pi}(S_t)$ OR

for each episode

for each time t

Agent observes S_t

Agent selects action A_t using π_θ

Envienment responds by transitioning from S_t to S_{t+1} and emitting reward R_t

$\delta_t = R_t + \gamma v^\pi(S_{t+1}) - v^\pi(S_t)$ → critic judging the actor.

$\forall i,\ \theta_i \leftarrow \theta_i + \alpha\, \delta_t \frac{\partial \ln(\pi_\theta(S_t, A_t))}{\partial \theta_i}$ → actor update

$\forall j,\ w_j \leftarrow w_j + \beta\, \delta_t \frac{\partial v_w(S_t)}{\partial w_j}$ → critic update

Run the episode

place ★ here

~~if $\sum_{t=0}^{\infty} \gamma^t R_t$ is big~~ $\sum_{k=0}^{\infty} \gamma^k R_{t+k}$ → R_t and S_{t+1} are all we have.

optional. (common)

for each time t

~~Make action A_t more likely in state S_t~~

~~$\forall i,\ \theta_i \leftarrow \theta_i + \alpha \frac{\partial \ln(\pi(S_t, A_t))}{\partial \theta_i}$~~ → How to change θ_i to increase $\pi_\theta(S_t, A_t)$

$\forall i,\ \theta_i \leftarrow \theta_i + \alpha \left(\sum_{k=0}^{\infty} \gamma^k R_{t+k}\right) \frac{\partial \ln(\pi_\theta(S_t, A_t))}{\partial \theta_i}$

replace w/ TD error, δ_t: $R_t + \gamma v^\pi(S_{t+1}) - v^\pi(S_t)$ (what happened after step t − what we expected at time t)

Actor-Critic

learn from the episode

~~if $\sum_{t=0}^{\infty} \gamma^t R_t$ is small~~

for each time t

Make action A_t less likely in state S_t

$\forall i,\ \theta_i \leftarrow \theta_i - \alpha \frac{\partial \ln(\pi_\theta(S_t, A_t))}{\partial \theta_i}$

~~cut all of this.~~

★ convert Eng → math

★ get rid of "if $\sum R_t$ is big"

★ A_t only responsible for rewards that happen after A_t.

★ move update into episode using TD error δ_t.

★ learn v^π

★ Move v^π update into episode using δ_t.

★ $v_w \approx v^\pi$

input	output
S_0	$\sum_{k=0}^{\infty} \gamma^k R_{0+k}$
S_1	$\sum_{k=0}^{\infty} \gamma^k R_{1+k}$
⋮	$R_t + \gamma v_w(S_{t+1})$

For each time t:

$\forall j \quad w_j \leftarrow w_j - \beta \delta_t \left(\sum_{k=0}^{\infty} \gamma^k R_{t+k} - v_w(S_t)\right) \cdots \frac{\partial v_w(S_t)}{\partial w_j}$

Supervised learning

x_i = input for i^{th} point

y_i = output/label for i^{th} point

n = number of points

$\hat{y}_i$ = agent's prediction of y_i from x_i.

non-parametric: NN
kNN
wkNN

parametric:

w_k = model parameters after k updates.

$f_{w_k}(x_i) = \hat{y}_i$

Linear parametric model:

$\phi_j(x_i)$ = j^{th} feature for input x_i. (basis function)

$$f_{w_k}(x_i) = \sum_j w_{kj}\,\phi_j(x_i)$$

$\ell(w_k)$ = loss function. (smaller = better w_k)

least squares loss:

$$\ell(w_k) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Other forms:

$$\ell(w_k) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\ell(w_k) = \frac{1}{2n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Hill Climbing: (descent)

~~sin(x)~~ is Lipschitz, $L = 1$

~~sin(x^2)~~ is not Lipschitz.

Gradient descent:

$$w_{k+1,j} = w_{k,j} - \alpha\, \partial \ell(w_k)/\partial w_{k,j}$$

α = step size.

α_k = step size for k^{th} update.

Remember all hyperparams, like when to stop!

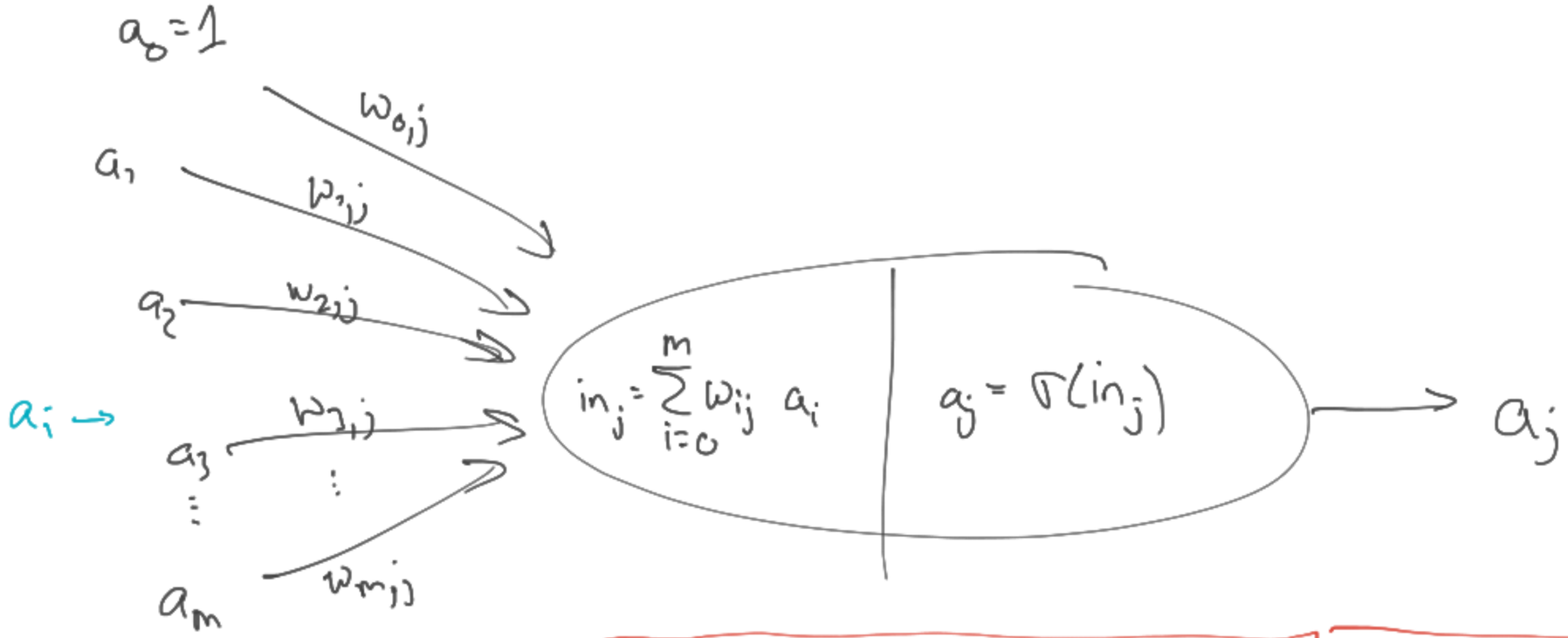
Convergence:

- Square summable, not summable

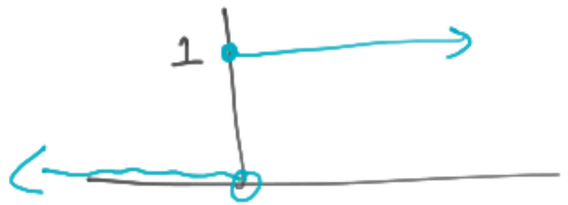
$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty \qquad \sum_{k=0}^{\infty} \alpha_k = \infty$$

Lipschitz continuity
→ If differentiable function, max slope is L.

Don't forget to normalize inputs!
Each feature
→ mean = 0
→ variance: 1



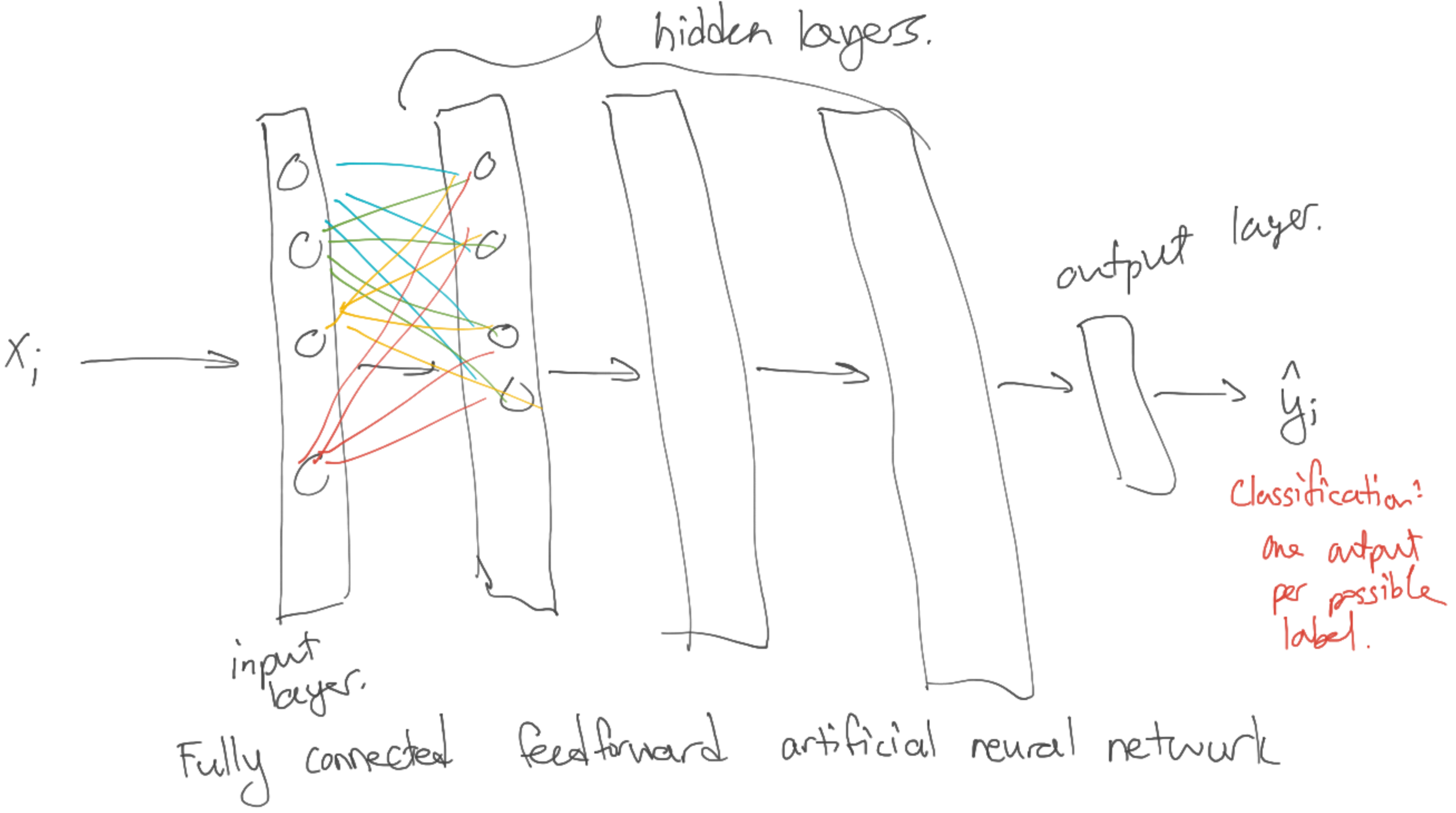
$$\sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



$\sigma(z)$ = sigmoid
→ logistic function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)\left(1 - \sigma(z)\right)$$



Fully connected feedforward artificial neural network

Backpropagation:

Old notation

$$\frac{\partial l(w_k)}{\partial w_{kj}} = \left(\quad \right) \frac{\partial f_{w_k}(x_i)}{\partial w_{kj}}$$

New notation

$$\frac{\partial f_w(x)}{\partial w_{ij}} = \left(\text{chain rule over, and over} \right)$$

i = prev layer
j = cur layer
k = next layer.

Other:

- Vanishing gradients
- over-fitting
- train, validation, test
- adaptive step sizes
- generalization bounds (safety)
  - Hoeffding's inequality applied to $\ell(w_k)$
  - using test data!
- fairness
- psych / neuro
- ethics
- philosophy of mind